XBeach Annual Report and Model Description

Dano Roelvink (P.I.), Ad Reniers,
Ap van Dongeren, Jaap van Thiel de Vries, Jamie Lescinski, Dirk-Jan Walstra

UNESCO-IHE Institute for Water Education (Contractor)
WL | Delft Hydraulics
Delft University of Technology

Contract no. N62558-06-C-2006

The Research reported in this document has been made possible through the support and sponsorship of the U.S. Government through its European Research Office of the U.S. Army. This report is intended only for the internal management use of the Contractor and U.S. Government.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT DATE	3. DATES COVERED *(From - To)*
01-03-2007	Interim report	September-November 2006

4. TITLE AND SUBTITLE

Modeling of
Hurricane Impacts,
Interim Report 2

5a. CONTRACT NUMBER

N62558-06-C-2006

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Roelvink, Dano
Reniers, Ad
van Dongeren, Ap
Van Thiel de Vries, Jaap
Lescinski, Jamie
Walstra, Dirk-Jan

5d. PROJECT NUMBER

5e. TASK NUMBER

0004

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

UNESCO-IHE INSTITUTE FOR WATER EDUCATION
PO BOX 3015
2601 DA DELFT
NETHERLANDS

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

FISC SIGONELLA
NAVAL REGIONAL CONTRACTING DET LONDON
SHORE/FLEET TEAM
BLOCK 2, WING 12, GOVT BLDGS LIME GROVE
RUISLIP MIDDLESEX HA4 8BX
UNITED KINGDOM

10. SPONSOR/MONITOR'S ACRONYM(S)

ERDC

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES

Prepared in collaboration with WL | Delft Hydraulics and Delft University of Technology

14. ABSTRACT

This first annual report describes ongoing development and validation of the XBeach model as part of the MORPHOS project and other activities over the period 1 March 2006-1 March 2007, and provides documentation and user manual for the present version of the model.

15. SUBJECT TERMS

Surf zone, swash, overwash, wave groups, wave propagation, morphology, dune erosion

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Prof. Dano Roelvink
unclassified	unclassified	unclassified	unclassified	54	19b. TELEPONE NUMBER (*Include area code*) +31 15 2151838

Contents

Abstract

The XBeach program contains a number of Fortran 90/95 routines for short wave propagation, nonstationary shallow water equations, sediment transport and continuity equations that can be coupled in various ways and are designed to cope with extreme conditions such as encountered during hurricanes. Since length scales are short in terms of wave lengths and supercritical flow frequently occurs, the numerical implementation is mainly first order upwind, which in combination with a staggered grid makes the model robust. The model scheme utilizes explicit schemes with an automatic time step based on Courant criterion, with output at fixed time intervals, which keeps the code simple and makes coupling and parallellization easier, while increasing stability.

The short wave propagation model contains a newly-developed time-dependent wave action balance solver, which solves the wave refraction and allows variation of wave action in x, y, time and over the directional space, and can be used to simulate the propagation and dissipation of wave groups. An added advantage to this set-up, compared to the existing surfbeat model, is that a separate wave model is not needed to predict the mean wave direction, and it allows different wave groups to travel in different directions. Through a variety of principle tests we are able to simulate surfbeats running up and over dunes. Full wave-current interaction in the short wave propagation is included. Roelvink (1993) wave dissipation model is implemented for use in the nonstationary wave energy balance (in other words, when the wave energy varies on the wave group timescale).

The Generalised Lagrangean Mean (GLM) approach was implemented to represent the depth-averaged undertow and its effect on bed shear stresses and sediment transport, cf. Reniers et al. (2004). The numerical scheme was updated, in line with Stelling and Duinmeijer method, to improve long-wave runup and backwash on the beach. The momentum-conserving form is applied, while retaining the simple first-order approach. The resulting scheme has been verified with the well-known Carrier and Greenspan test.

Soulsby – Van Rijn transport formulations have been included, which solves the 2DH advection-diffusion equation and produces total transport vectors, which can be used to update the bathymetry. The pickup function follows Reniers et al (2004) was implemented. An avalanching routine was implemented with separate criteria for critical slope at wet or dry points.

The model has been validated against a number of analytical and laboratory tests, both hydrodynamic and morphodynamic.

1 Introduction

This report is the annual report of the project 'Modeling of Hurricane Impacts', contract no. N62558-06-C-2006, which was granted by the US Army Corps of Engineers, Engineer Research and Development Center (ERDC), European Research Office and administered by FISC SIGONELLA, NAVAL REGIONAL CONTRACTING DET LONDON, SHORE/FLEET TEAM.

The project is being carried out by Prof. Dano Roelvink of UNESCO-IHE (Principal Investigator), Dr. Ad Reniers and Jaap van Thiel de Vries of Delft University of Technology and Dr. Ap van Dongeren, Dirk-Jan Walstra and Jamie Lescinski of WL | Delft Hydraulics.

1.1 Objective

The main objective of the XBeach model is to provide a robust and flexible environment in which to test morphological modeling concepts for the case of dune erosion, overwashing and breaching. The top priority is to provide numerical stability; first order accuracy is accepted since there is a need for small space steps and time steps anyway, to represent the strong gradients in space and time in the nearshore and swash zone. Because of the many shock-like features in both hydrodynamics and morphodynamics we choose upwind schematizations as a means to avoid numerical oscillations which can be deadly in shallow areas.

The modeling environment should be flexible and the code easy to comprehend and concise; therefore we have adapted the Matlab environment as development environment; and converted to Fortran 90/95 at a later stage. As of now, the future development is in the Fortran 90/95 environment.

1.2 Context

The XBeach model can be used as stand-alone model for small-scale (project-scale) coastal applications, but will also be used within the Morphos model system, where it will be driven by boundary conditions provided by the wind, wave and surge models and its main output to be transferred back will be the time-varying bathymetry and possibly discharges over breached barrier island sections.

1.3 Functionalities

The code has the following functionalities:

- Depth-averaged shallow water equations including time-varying wave forcing terms; combination of sub- and supercritical flows;
- Time-varying wave action balance including refraction, shoaling, current refraction and wave breaking;
- Roller model, including breaker delay
- Wave amplitude effects on wave celerity;
- Depth-averaged advection-diffusion equation to solve suspended transport;
- Bed updating algorithm including possibility of avalanching;
- Possibility to extend to parallel multi-domain version;
- Numerical scheme in line with Stelling and Duinmeijer method, to improve long-wave runup and backwash on the beach. The momentum-conserving form is applied, while retaining the simple first-order approach. The resulting scheme has been tested against the well-known Carrier and Greenspan test.
- Generalised Lagrangean Mean (GLM) approach to represent the depth-averaged undertow and its effect on bed shear stresses and sediment transport, cf. Reniers et al. (2004)
- Roelvink (1993) wave dissipation model for use in the nonstationary wave energy balance (in other words, when the wave energy varies on the wave group timescale)
- Soulsby – Van Rijn transport formulations, cf Reniers et al (2004).
- Automatic time step based on Courant criterion, with output at fixed time intervals.
- Avalanching mechanism, with separate criteria for critical slope at wet or dry points.

1.4 Outline of the report

In the following report we will detail the model development, activities and results. Chapter 2 provides a description of the XBeach structure, as well as an overview of siginificant attributes of the program. Chapter 3 contains an update of all formulations used and their numerical schematization. In Chapter 4, a series of validation tests are described, ranging from analytical experiments to check the numerical behavior, to large-scale dune erosion (for an experiment in a wave flume and a field study) simulations where mass avalanches were observed. Chapter 5 provides instructions on how to run the model, with a detailed input description. Chapter 6 offers a small discussion of how it is intended to distribute and maintain XBeach. In Chapter 7 we draw conclusions and sketch our plans for the coming period.

2 Description of program structure

The program XBeach consists of a main *Fortran 90 script*, xbeach.f90, and a number of *subroutines* that operate on two *structures*:

- par – this contains general input parameters
- s - this contains all the arrays for a given computational domain

For a single-domain run, one structure s is passed between flow, wave, sediment and bed update solvers, which extract the arrays they need from the structure elements to local variables, do their thing and pass the results back to the relevant structure elements. This makes the overall program clear, prevents long parameter lists and makes it easy to add input variables or arrays where needed.

For multi-domain runs, one can define multiple instantiations of the structure s which are passed to the same functions; an additional function is needed to pass the boundary information between the domains back and forth. We have carried out a simple test of this principle, without actually implementing a multi-processor version, which confirms that the data structure can handle this case.

In the Table 1 we will outline the various functions and their purposes. The main program xbeach.f90 is reproduced in Table 2 below. (Note: minor cleaning up has to be done in this routine, such that only subroutine calls are carried out here).

Table 1. Overview of Fortran 90 subroutine calls by xbeach.f90

Function call	Purpose
call wave_input(par)	Creates elements of structure par containing wave input parameters
call flow_input(par)	Adds elements of structure par containing flow input parameters
call sed_input(par)	Adds elements of structure par containing sediment input parameters
call grid_bathy(s)	Creates grid and bathymetry and stores them in structure s
call wave_dist(s,par)	Creates initial directional spectrum at sea boundary
call wave_init (s,par)	Initialises arrays (elements of s) for wave computations
call flow_init (s,par)	Initialises arrays (elements of s) for flow computations
call sed_init (s,par)	Initialises arrays (elements of s) for sediment computations
call wave_bc (s,par,it)	Wave boundary conditions update, each timestep
call flow_bc (s,par,it)	Flow boundary conditions update, each timestep
call wave_timestep(s,par)	Carries out one wave timestep
call flow_timestep (s,par)	Carries out one flow timestep
call transus(s,par)	Carries out one suspended transport timestep
call bed_update(s,par)	Carries out one bed level update timestep
call output(it,s,par)	Performs output

```
program xbeach
use params
use spaceparams

IMPLICIT NONE

type(parameters)        ::par
type(spacepars)         ::s

integer                          ::it

! General input per module
call wave_input(par)
call flow_input(par)
call sed_input(par)

! Grid and bathymetry
call grid_bathy(s)

! Directional distribution wave energy
call wave_dist(s,par)

! Initialisations
call wave_init (s,par)
call flow_init (s,par)
call sed_init (s,par)

it=0

do while(par%t<=par%tstop)
   ! Wave boundary conditions
   call wave_bc (s,par,it);
   ! Flow boundary conditions
   call flow_bc (s,par,it);
   ! Wave timestep
   call wave_timestep(s,par)
   ! Flow timestep
   call flow_timestep (s,par)
   ! Suspended transport
   call transus(s,par)
   ! Bed level update
   call bed_update(s,par)
   ! Output
   call output(it,s,par)
enddo
end program
```

Table 2 Main routine xbeach.m

2.1 Overview of updates to Fortran code

In the first three months thru May 2006, a first (Matlab) version of XBeach was constructed with the following functionality:

- Depth-averaged shallow water equations including time-varying wave forcing terms; combination of sub- and supercritical flows;
- Time-varying wave action balance including refraction, shoaling, current refraction and wave breaking;
- Wave amplitude effects on wave celerity;
- Depth-averaged advection-diffusion equation to solve suspended transport;
- Bed updating algorithm including possibility of avalanching;
- Possibility to extend to parallel multi-domain version;

The following improvements and additions have been implemented and tested in the period June-August 2006:

- Improvement of numerical scheme in line with Stelling and Duinmeijer method, to improve long-wave runup and backwash on the beach. The momentum-conserving form is applied, while retaining the simple first-order approach. The resulting scheme has been tested against the well-known Carrier and Greenspan test.
- Implementation of the Generalised Lagrangean Mean (GLM) approach to represent the depth-averaged undertow and its effect on bed shear stresses and sediment transport, cf. Reniers et al. (2004)
- Implementation of Roelvink (1993) wave dissipation model for use in the nonstationary wave energy balance (in other words, when the wave energy varies on the wave group timescale)
- Implementation of Soulsby – Van Rijn transport formulations, cf Reniers et al (2004).
- Automatic time step based on Courant criterion, with output at fixed time intervals.
- Improvement of avalanching mechanism, with separate criteria for critical slope at wet or dry points.

In the period September-November 2006, the following updates were reported:

- Conversion to Fortran 90/95
- Implementation of weakly reflective boundary conditions cf. Van Dongeren and Svendsen
- Adding generic input routines, allowing all input to be specified by simple ASCII input files, which are either free-format bulk data (bathymetry, time series) or in a keyword-based *.ini* file.

In the most recent period we implemented the roller model and breaker delay according to Reniers et al. (2004) and carried out a large number of validation tests, where all important tests were rerun with a unified code, in which all developments carried out were merged.

3 Model formulations

3.1 Coordinate system

XBeach uses a coordinate system where the computational x-axis is always oriented towards the coast, approximately perpendicular to the coastline, and the y-axis is alongshore. This coordinate system is defined relative to world coordinates (xw,yw) through the origin (xori,yori) and the orientation alfa, defined counter-clockwise w.r.t. the xw-axis (East).

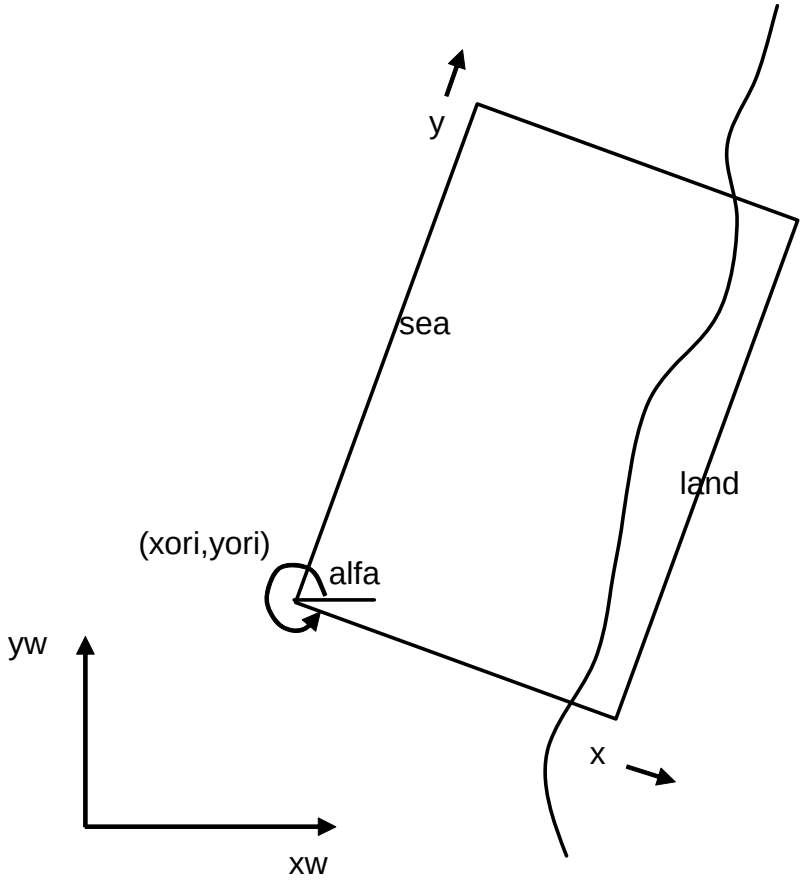

Figure 3-1 Coordinate system

3.2 Grid Setup

The grid applied is a staggered grid, where the bed levels, water levels, water depths and concentrations are defined in cell centers, and velocities and sediment transports are defined in u- and v-points, viz. at the cell interfaces. In the wave energy balance, the energy, roller energy and radiation stress are defined at the cell centers, whereas the radiation stress gradients are defined at u- and v-points.

Velocities at the u- and v-points are denoted by uu and vv respectively; velocities u and v at the cell centers are obtained by interpolation and are for output purpose only. The water level, zs, and the bed level, zb, are both defined positive upward.

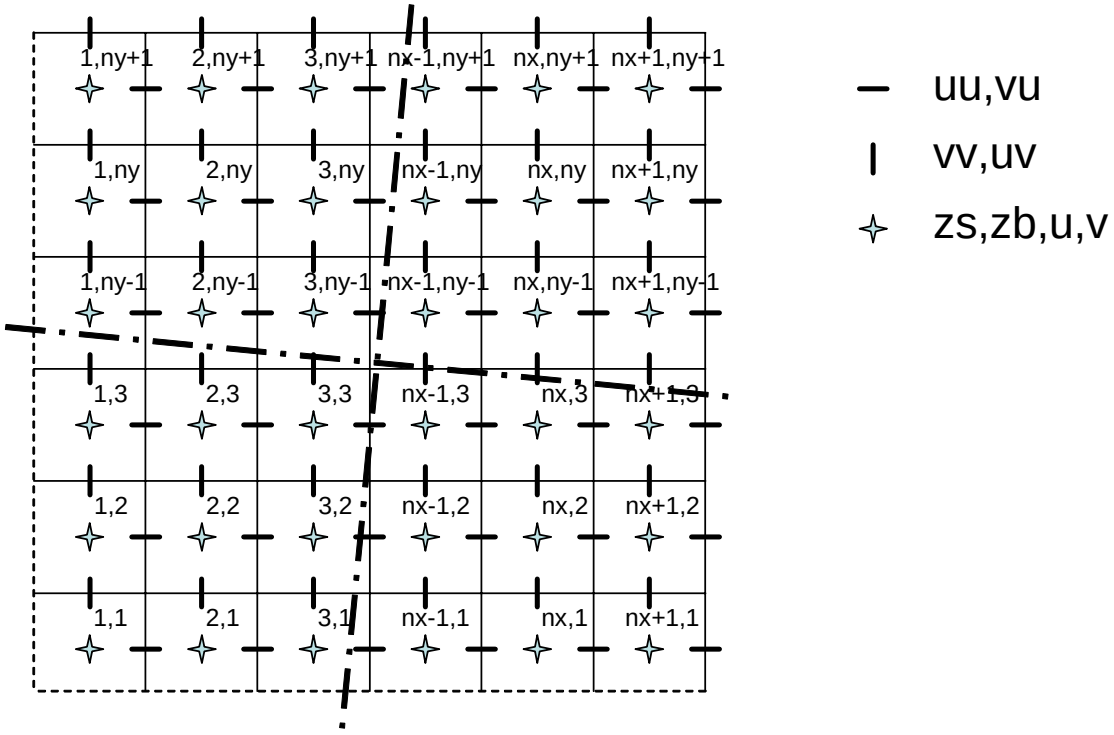

Figure 3-2 Staggered grid

3.3 Wave action equation solver

The wave forcing in the shallow water momentum equation is obtained from a time dependent version of the wave action balance equation. Similar to Delft University's HISWA model, the directional distribution of the action density is taken into account whereas the frequency spectrum is represented by a single mean frequency. The wave action balance is then given by:

$$\frac{\partial A}{\partial t}+\frac{\partial c_x A}{\partial x}+\frac{\partial c_y A}{\partial y}+\frac{\partial c_\theta A}{\partial \theta}=-\frac{D}{\sigma} \tag{3.1}$$

with the wave action:

$$A(x,y,\theta)=\frac{S_w(x,y,\theta)}{\sigma(x,y)} \tag{3.2}$$

where S_w represents the wave energy in each directional bin and σ the intrinsic wave frequency. The wave action propagation speeds in x- and y-direction are given by:

$$\begin{aligned} c_x(x,y,\theta) &= c_g(x,y)\bullet\cos(\theta) + u(x,y) \\ c_y(x,y,\theta) &= c_g(x,y)\bullet\sin(\theta) + v(x,y) \end{aligned} \tag{3.3}$$

where θ represents the angle of incidence with respect to the x-axis. The propagation speed in θ-space is obtained from:

$$\begin{aligned} c_\theta(x,y,\theta) = \frac{\sigma}{\sinh 2kh}\left(\frac{\partial h}{\partial x}\sin\theta - \frac{\partial h}{\partial y}\cos\theta\right) + \cos\theta\left(\sin\theta\frac{\partial u}{\partial x} - \cos\theta\frac{\partial u}{\partial y}\right) + \\ + \sin\theta\left(\sin\theta\frac{\partial v}{\partial x} - \cos\theta\frac{\partial v}{\partial y}\right) \end{aligned} \tag{3.4}$$

taking into account bottom refraction (first term on the RHS) and current refraction (last two terms on the RHS). The wave number k is obtained from the eikonal equations:

$$\begin{aligned} \frac{\partial k_x}{\partial t} + \frac{\partial \omega}{\partial x} = 0 \\ \frac{\partial k_y}{\partial t} + \frac{\partial \omega}{\partial y} = 0 \end{aligned} \tag{3.5}$$

where the subscripts refer to the direction of the wave vector components and ω represents the absolute radial frequency. The wave number is the obtained from:

$$k = \sqrt{k_x^2 + k_y^2} \tag{3.6}$$

The absolute radial frequency is given by:

$$\omega = \sigma + \vec{k}\bullet\vec{u} \tag{3.7}$$

and the intrinsic frequency is obtained from the linear dispersion relation:

$$\sigma = \sqrt{gk\tanh kh} \tag{3.8}$$

The group velocity is obtained from linear wave theory:

$$c_g = nc = \left(\frac{1}{2} + \frac{kh}{\sinh 2kh}\right)\frac{\sigma}{k} \tag{3.9}$$

This concludes the advection of wave action. The wave energy dissipation due to wave breaking is modelled according to Baldock et al. [1998]:

$$\bar{D} = \frac{1}{4}\alpha Q_b \rho g f_m \left(H_b^2 + H_{rms}^2 \right) \tag{3.10}$$

with $\alpha = O(1)$ and f_m representing the mean intrinsic frequency. The fraction of breaking waves is given by:

$$Q_b = \exp\left[-\left(\frac{H_b^2}{H_{rms}^2} \right) \right] \tag{3.11}$$

where the breaking wave height is:

$$H_b = \frac{0.88}{k} \tanh\left[\frac{\gamma k h}{0.88} \right] \tag{3.12}$$

and γ is a calibration parameter. The root mean square wave height is obtained from:

$$H_{rms} = \sqrt{\frac{8\int S_w(x,y,\theta) d\theta}{\rho g}} = \sqrt{\frac{8E_w}{\rho g}} \tag{3.13}$$

Next the total wave dissipation, $\bar{D}$, is distributed proportionally over the wave directions:

$$D(x,y,\theta) = \frac{S_w(x,y,\theta)}{E_w(x,y)} \bar{D} \tag{3.14}$$

This closes the set of equations for the wave action balance. Given the spatial distribution of the wave action and therefore wave energy the wave forcing can be calculated utilizing the radiation stress tensor:

$$\begin{aligned} F_x &= -\left(\frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{xy}}{\partial y} \right) \\ F_y &= -\left(\frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y} \right) \end{aligned} \tag{3.15}$$

And:

$$\begin{aligned} S_{xx} &= \int \left(\frac{c_g}{c} \left(1 + \cos^2\theta \right) - \frac{1}{2} \right) S_w d\theta \\ S_{xy} = S_{yx} &= \int \sin\theta \cos\theta \left(\frac{c_g}{c} S_w \right) d\theta \\ S_{yy} &= \int \left(\frac{c_g}{c} \left(1 + \sin^2\theta \right) - \frac{1}{2} \right) S_w d\theta \end{aligned} \tag{3.16}$$

We use an up-wind schematisation to solve the wave action balance. The wave action is given at the same points at the water level. The advection of wave action is then discretized as follows:

$$\begin{aligned}\frac{\partial c_x^n A^n}{\partial x}(\mathrm{i,j,k}) &= \frac{c^n_{x,i,j,k}A^n_{i,j,k} - c^n_{x,i-1,j,k}A^n_{i-1,j,k}}{x_{\mathrm{i,j}} - x_{\mathrm{i\text{-}1,j}}} \quad , c^n_{x,i,j,k} > 0 \\ \frac{\partial c_x^n A^n}{\partial x}(\mathrm{i,j,k}) &= \frac{c^n_{x,i+1,j,k}A^n_{i+1,j,k} - c^n_{x,i,j,k}A^n_{i,j,k}}{x_{\mathrm{i+1,j}} - x_{\mathrm{i,j}}} \quad , c^n_{x,i,j,k} < 0\end{aligned} \tag{3.17}$$

$$\begin{aligned}\frac{\partial c_y^n A^n}{\partial y}(\mathrm{i,j,k}) &= \frac{c^n_{y,i,j,k}A^n_{i,j,k} - c^n_{y,i,j-1,k}A^n_{i,j-1,k}}{y_{\mathrm{i,j}} - y_{\mathrm{i,j\text{-}1}}} \quad , c^n_{y,i,j,k} > 0 \\ \frac{\partial c_y^n A^n}{\partial y}(\mathrm{i,j,k}) &= \frac{c^n_{y,i,j+1,k}A^n_{i,j+1,k} - c^n_{y,i,j,k}A^n_{i,j,k}}{y_{\mathrm{i,j+1}} - y_{\mathrm{i,j}}} \quad , c^n_{y,i,j,k} < 0\end{aligned} \tag{3.18}$$

$$\begin{aligned}\frac{\partial c_\theta^n A^n}{\partial \theta}(\mathrm{i,j,k}) &= \frac{c^n_{\theta,i,j,k}A^n_{i,j,k} - c^n_{\theta,i,j,k-1}A^n_{i,j,k-1}}{\theta_{\mathrm{i,j,k}} - \theta_{\mathrm{i,j,k\text{-}1}}} \quad , c^n_{\theta,i,j,k} > 0 \\ \frac{\partial c_\theta^n A^n}{\partial \theta}(\mathrm{i,j,k}) &= \frac{c^n_{\theta,i,j,k+1}A^n_{i,j,k+1} - c^n_{\theta,i,j,k}A^n_{i,j,k}}{\theta_{\mathrm{i,j,k+1}} - \theta_{\mathrm{i,j,k}}} \quad , c^n_{\theta,i,j,k} < 0\end{aligned} \tag{3.19}$$

Similar for the wave action balance:

$$\frac{A^{n+1}_{i,j,k} - A^n_{i,j,k}}{\Delta t} = -\frac{\partial c_x^n A^n}{\partial x}_{i,j,k} - \frac{\partial c_y^n A^n}{\partial y}_{i,j,k} - \frac{\partial c_\theta^n A^n}{\partial \theta}_{i,j,k} - \frac{D}{\sigma}_{i,j,k} \tag{3.20}$$

which yields the wave energy at the new time level.

3.4 Roller energy equation solver

The roller energy balance is coupled to the wave action/energy balance where dissipation of wave energy serves as a source term for the roller energy balance. Similar to the wave action the directional distribution of the roller energy is taken into account whereas the frequency spectrum is represented by a single mean frequency. The roller energy balance is then given by:

$$\frac{\partial S_r}{\partial t} + \frac{\partial c_x S_r}{\partial x} + \frac{\partial c_y S_r}{\partial y} + \frac{\partial c_\theta S_r}{\partial \theta} = -D_r + D_w \tag{3.21}$$

with the roller energy:

$S_r(x, y, \theta)$

representing the roller energy in each directional bin. The roller energy propagation speeds in x- and y-direction are given by:

$$\begin{aligned} c_x(x,y,\theta) &= c(x,y)\bullet\cos(\theta) + u(x,y) \\ c_y(x,y,\theta) &= c(x,y)\bullet\sin(\theta) + v(x,y) \end{aligned} \tag{3.22}$$

where θ represents the angle of incidence with respect to the x-axis. The propagation speed in θ-space is obtained from:

$$\begin{aligned} c_\theta(x,y,\theta) = \frac{\sigma}{\sinh 2kh}\left(\frac{\partial h}{\partial x}\sin\theta - \frac{\partial h}{\partial y}\cos\theta\right) + \\ + \cos\theta\left(\sin\theta\frac{\partial u}{\partial x} - \cos\theta\frac{\partial u}{\partial y}\right) + \sin\theta\left(\sin\theta\frac{\partial v}{\partial x} - \cos\theta\frac{\partial v}{\partial y}\right) \end{aligned} \tag{3.23}$$

taking into account bottom refraction (first term on the RHS) and current refraction (last two terms on the RHS). Hence, we are assuming that the waves and rollers propagate in the same direction. The phase velocity is obtained from linear wave theory:

$$c = \frac{\sigma}{k} \tag{3.24}$$

which concludes the advection of roller energy. The roller energy dissipation is given by (Deigaard, 1993):

$$\overline{D}_r = c\tau_r \tag{3.25}$$

with τ_r representing the shear stress induced by the roller at the surface, which is expressed by (Svendsen, 1984):

$$\tau_r = \frac{\rho g R}{L}\beta_r \tag{3.26}$$

where *R* represents the roller area and β_r is the slope of the breaking wave. The roller area is related to the roller energy trough:

$$E_r = \frac{1}{2}\frac{\rho R c^2}{L} \tag{3.27}$$

Next the total wave dissipation, $\overline{D}_r$, is distributed proportionally over the wave directions:

$$D_r(x,y,\theta) = \frac{S_r(x,y,\theta)}{E_r(x,y)}\overline{D}_r \tag{3.28}$$

Similarly, the source term is obtained from the wave action/energy balance:

$$D_w(x,y,\theta) = \frac{S_w(x,y,\theta)}{E_w(x,y)}\overline{D} \tag{3.29}$$

This closes the set of equations for the roller energy balance. The roller also affects the wave forcing and has therefore to be included in the radiation stress terms:

$$\begin{aligned} S_{xx,r} &= \int \cos^2\theta S_r d\theta \\ S_{xy,r} &= S_{yx,r} = \int \sin\theta\cos\theta S_r d\theta \\ S_{yy,r} &= \int \sin^2\theta S_w d\theta \end{aligned} \tag{3.30}$$

These roller radiation stress contributions are added to the wave-induced radiation stresses. Similar to the solution of the wave action equations we use an up-wind schematisation to solve the roller energy balance.

3.5 Shallow water equations solver

Shallow water equations, neglecting Coriolis and horizontal diffusion terms, and (grey terms), for the moment, wind shear stress:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{\tau_{sx}}{\rho h} - \frac{\tau_{bx}}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h} \tag{3.31}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = +\frac{\tau_{sy}}{\rho h} - \frac{\tau_{by}}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \tag{3.32}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0 \tag{3.33}$$

Here, *h* is the water depth, *u*, *v* are velocities in *x* and and *y* direction, τ_{bx}, τ_{by} are the bed shear stresses, g is the acceleration of gravity, η is the water level and F_x, F_y are the wave-induced stresses.

We apply an upwind schematisation, since the horizontal scale of the problem is limited and such a scheme deals with shocks in a natural way.

We apply a staggered grid, where bed levels and water levels are defined in the centre of cells, and velocity components at the cell interfaces.

If *nx,ny* are the number of cells in both directions, the water level points are numbered from 1 to *nx+1* and from 1 to *ny+1*.

The water level gradients are computed at the cell interfaces and are given by:

$$\frac{\partial \eta}{\partial x}(\mathrm{i,j}) = \frac{\eta_{\mathrm{i+1,j}} - \eta_{\mathrm{i,j}}}{x_{\mathrm{i+1,j}} - x_{\mathrm{i,j}}} \tag{3.34}$$

$$\frac{\partial \eta}{\partial y}(\mathrm{i,j}) = \frac{\eta_{\mathrm{i,j+1}} - \eta_{\mathrm{i,j}}}{x_{\mathrm{i,j+1}} - x_{\mathrm{i,j}}} \tag{3.35}$$

For computing the shear stresses at the cell interfaces we need the velocity magnitudes at these interfaces. These are composed by combining the normal velocity component at the interface and the average of the 4 adjacent tangential components:

$$v_{u,i,j} = \frac{1}{4}(v_{i,j-1} + v_{i,j} + v_{i+1,j-1} + v_{i+1,j})$$
$$u_{v,i,j} = \frac{1}{4}(u_{i-1,j} + u_{i,j} + u_{i-1,j+1} + u_{i,j+1}) \quad (3.36)$$

The water depth in each cell is computed as:

$$h_{i,j} = \eta_{i,j} - z_{b,i,j} \quad (3.37)$$

For the depth at cell interfaces, following Stelling and Duinmeijer (2003) we distinguish between the depth used in the continuity equation and that used in the momentum equation. The depth at the interfaces *for the continuity equation* is taken as the upwind depth in case the velocity is greater than a minimum velocity, or the maximum water level minus the maximum bed level in case the velocity is less than this minimum velocity:

$$\begin{aligned} h_{u,i,j} &= h_{i,j} && , u_{i,j} > u_{\min} \\ h_{u,i,j} &= h_{i+1,j} && , u_{i,j} < -u_{\min} \\ h_{u,i,j} &= \max(z_{s,i,j}, z_{s,i+1,j}) - \max(z_{b,i,j}, z_{b,i+1,j}) && , |u_{i,j}| < u_{\min} \end{aligned} \quad (3.38)$$

$$\begin{aligned} h_{v,i,j} &= h_{i,j} && , v_{i,j} > v_{\min} \\ h_{v,i,j} &= h_{i,j+1} && , v_{i,j} < -v_{\min} \\ h_{v,i,j} &= \max(z_{s,i,j}, z_{s,i,j+1}) - \max(z_{b,i,j}, z_{b,i,j+1}) && , |v_{i,j}| < v_{\min} \end{aligned} \quad (3.39)$$

For the depth *in the momentum balance* we take the average depth between the cell centers:

$$h_{mu,i,j} = \frac{1}{2}(h_{i,j} + h_{i+1,j}) \quad (3.40)$$

$$h_{mv,i,j} = \frac{1}{2}(h_{i,j} + h_{i,j+1}), \quad (3.41)$$

The advection terms in x-direction are approximated as follows:

$$\begin{aligned} u\frac{\partial u}{\partial x}^{n}_{i,j} &= \frac{1}{2}\frac{h_{u,i,j}u_{i,j} + h_{u,i-1,j}u_{i-1,j}}{h_{mu,i,j}}\frac{u^n_{i,j} - u^n_{i-1,j}}{x^n_{i,j} - x^n_{i-1,j}} && , u^n_{i,j} > 0 \\ u\frac{\partial u}{\partial x}^{n}_{i,j} &= \frac{1}{2}\frac{h_{u,i,j}u_{i,j} + h_{u,i+1,j}u_{i+1,j}}{h_{mu,i,j}}\frac{u^n_{i+1,j} - u^n_{i,j}}{x^n_{i+1,j} - x^n_{i,j}} && , u^n_{i,j} < 0 \end{aligned} \quad (3.42)$$

$$v\frac{\partial u}{\partial y}^{n}_{i,j} = v^{n}_{u,i,j}\frac{u^n_{i,j+1} - u^n_{i,j-1}}{y^n_{i,j+1} - y^n_{i,j-1}} \quad (3.43)$$

The advection terms in y-direction are approximated as follows:

$$v\frac{\partial v^n}{\partial y_{i,j}} = \frac{1}{2}\frac{h^n_{v,i,j}v^n_{i,j} + h^n_{v,i,j-1}v^n_{i,j-1}}{h^n_{mv,i,j}}\frac{v^n_{i,j} - v^n_{i,j-1}}{y^n_{i,j} - y^n_{i,j-1}} \qquad ,v^n_{i,j} > 0$$
$$v\frac{\partial v^n}{\partial y_{i,j}} = \frac{1}{2}\frac{h^n_{v,i,j}v^n_{i,j} + h^n_{v,i,j+1}v^n_{i,j+1}}{h^n_{mv,i,j}}\frac{v^n_{i,j+1} - v^n_{i,j}}{y^n_{i,j+1} - y^n_{i,j}} \qquad ,v^n_{i,j} < 0 \tag{3.44}$$

$$u\frac{\partial v^n}{\partial x_{i,j}} = u^n_{v,i,j}\frac{v^n_{i+1,j} - v^n_{i+1,j}}{x^n_{i+1,j} - x^n_{i+1,j}} \tag{3.45}$$

The momentum equation is discretized as follows:

$$\frac{u^{n+1}_{i,j} - u^n_{i,j}}{\Delta t} = -u\frac{\partial u^n}{\partial x_{i,j}} - v\frac{\partial u^n}{\partial y_{i,j}} - \frac{g\,u^n_{i,j}\sqrt{u^{n\,2}_{i,j} + v^{n\,2}_{u,i,j}}}{h^n_{u,i,j}C^2} - g\frac{\eta^n_{i+1,j} - \eta^n_{i,j}}{x_{i+1,j} - x_{i,j}} + \frac{F_{x,i,j}}{\rho h_{u,i,j}} \tag{3.46}$$

$$\frac{v^{n+1}_{i,j} - v^n_{i,j}}{\Delta t} = -v\frac{\partial v^n}{\partial y_{i,j}} - u\frac{\partial v^n}{\partial x_{i,j}} - \frac{g\,v^n_{i,j}\sqrt{u^{n\,2}_{v,i,j} + v^{n\,2}_{i,j}}}{h^n_{v,i,j}C^2} - g\frac{\eta_{i,j+1} - \eta_{i,j}}{y_{i,j+1} - y_{i,j}} + \frac{F_{y,i,j}}{\rho h_{v,i,j}} \tag{3.47}$$

From this, the velocities at the new time step level are computed. The water level is then updated by:

$$\frac{\eta^{n+1}_{i,j} - \eta^n_{i,j}}{\Delta t} = -\frac{u^{n+1}_{i,j}h^n_{i,j} - u^{n+1}_{i-1,j}h^n_{i-1,j}}{x_{u,i,j} - x_{u,i-1,j}} - \frac{v^{n+1}_{i,j}h_{i,j} - v^{n+1}_{i,j-1}h^n_{i,j-1}}{y_{v,i,j} - y_{v,i,j-1}} \tag{3.48}$$

Generalized Lagrangian Mean formulation

To account for the wave induced mass-flux and the subsequent (return) flow the shallow water equations are cast into a Generalized Lagrangian Mean (GLM) formulation (Walstra et al, 2000). To that end the Eulerian shallow water velocity u^E is replaced with its lagrangian equivalent, u^L:

$$u^L = u^E + u^S \quad and \quad v^L = v^E + v^S \tag{3.49}$$

and u^S , v^S represents the Stokes drift in x- and y-direction respectively (Phillips, 1977):

$$u^S = \frac{E_w\cos\theta}{\rho hc} \quad and \quad v^S = \frac{E_w\sin\theta}{\rho hc} \tag{3.50}$$

where the wave-group varying short wave energy and direction are obtained from the wave-action balance. The resulting GLM-momentum equations are given by:

$$\frac{\partial u^L}{\partial t} + u^L \frac{\partial u^L}{\partial x} + v^L \frac{\partial u^L}{\partial y} = -\frac{\tau_{bx}^E}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h}$$
$$\frac{\partial v^L}{\partial t} + u^L \frac{\partial v^L}{\partial x} + v^L \frac{\partial v^L}{\partial y} = -\frac{\tau_{by}^E}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \quad (3.51)$$

for the x- and y-direction respectively. This operation shows that the GLM equations for the depth-averaged flow are very similar to the previously described Eulerian formulation, with the exception of the bottom shear stress terms that are calculated with the Eulerian velocities as experienced by the bed:

$$u^E = u^L - u^S \quad and \quad v^E = v^L - v^S \quad (3.52)$$

and not with the GLM velocities. Also, the boundary condition for the flow computations has to be expressed in functions of (u^L , v^L) and not (u^E, v^E).

3.6 Sediment transport

Advection-diffusion scheme

The sediment transport is modeled with a depth-averaged advection diffusion equation [Gallapatti, 1983]:

$$\frac{\partial hC}{\partial t} + \frac{\partial hCu^E}{\partial x} + \frac{\partial hCv^E}{\partial y} + \frac{\partial}{\partial x}\left[D_h h \frac{\partial C}{\partial x}\right] + \frac{\partial}{\partial y}\left[D_h h \frac{\partial C}{\partial y}\right] = \frac{hC_{eq} - hC}{T_s} \quad (3.53)$$

where C represents the depth-averaged sediment concentration which varies on the infragravity time scale. The entrainment of the sediment is represented by an adaptation time T_s, given by a simple approximation based on the local water depth, h, and sediment fall velocity w_s:

$$T_s = \max\left(0.05\frac{h}{w_s}, 0.2\right) s \quad (3.54)$$

where a small value of T corresponds to nearly instantaneous sediment response. The entrainment or deposition of sediment is determined by the mismatch between the actual sediment concentration and the equilibrium concentration, C_{eq}, thus representing the source term in the sediment transport equation.

The differential equations for the advection diffusion of sediment is solved with finite differences using the first order up-wind scheme discussed earlier with the water depths at the old time level and the corresponding velocities at the new time level. The horizontal x-advection is then given by:

$$\left(\frac{\partial hCu^E}{\partial x}\right)_{i,j} = \frac{\left(h^n C^n u^{E,n+1}\right)_{i,j} - \left(h^n C^n u^{E,n+1}\right)_{i-1,j}}{x_{i,j} - x_{i-1,j}} \qquad ,u_{i,j}^{E,n+1} > 0$$

$$\left(\frac{\partial hCu^E}{\partial x}\right)_{i,j} = \frac{\left(h^n C^n u^{E,n+1}\right)_{i+1,j} - \left(h^n C^n u^{E,n+1}\right)_{i,j}}{x_{i+1,j} - x_{i,j}} \qquad ,u_{i,j}^{E,n+1} < 0 \qquad (3.55)$$

a similar expression for the horizontal advection in the y-direction:

$$\left(\frac{\partial hCv^E}{\partial y}\right)_{i,j} = \frac{\left(h^n C^n v^{E,n+1}\right)_{i,j} - \left(h^n C^n v^{E,n+1}\right)_{i-1,j}}{y_{i,j} - y_{i,j-1}} \qquad ,v_{i,j}^{E,n+1} > 0$$

$$\left(\frac{\partial hCv^E}{\partial y}\right)_{i,j} = \frac{\left(h^n C^n v^{E,n+1}\right)_{i,j+1} - \left(h^n C^n v^{E,n+1}\right)_{i,j}}{y_{i,j+1} - y_{i,j}} \qquad ,v_{i,j}^{E,n+1} < 0 \qquad (3.56)$$

The horizontal diffusion is evaluated at the old time level *n* and approximated by:

$$\left(\frac{\partial}{\partial x}\left(D_H h \frac{\partial C}{\partial x}\right)\right)_{i,j} = \frac{\left(D_H h C_{\partial x}\right)_{i+1,j} - \left(D_H h C_{\partial x}\right)_{i,j}}{x_{i+1,j} - x_{i,j}} \qquad (3.57)$$

Where the cross-shore gradient in the sediment concentration is given by:

$$C_{\partial x} = \left(\frac{\partial C}{\partial x}\right)_{i,j} = \frac{C_{i+1,j} - C_{i,j}}{x_{i+1,j} - x_{i,j}} \qquad (3.58)$$

And similarly for the y-direction:

$$\left(\frac{\partial}{\partial y}\left(D_H h \frac{\partial C}{\partial y}\right)\right)_{i,j} = \frac{\left(D_H h C_{\partial y}\right)_{i,j+1} - \left(D_H h C_{\partial y}\right)_{i,j}}{y_{i,j+1} - y_{i,j}} \qquad ,v_{i,j}^{E} < 0 \qquad (3.59)$$

Where the along-shore gradient in the sediment concentration, C_y , is given by:

$$\left(\frac{\partial C}{\partial y}\right)_{i,j} = \frac{C_{i,j+1} - C_{i,j}}{y_{i,j+1} - y_{i,j}} \qquad (3.60)$$

The time up-date of the sediment concentration is then given by:

$$\frac{h_{i,j}^{n+1}C_{i,j}^{n+1}-h_{i,j}^{n}C_{i,j}^{n}}{\Delta t}+\left[\frac{\partial hCu^{E}}{\partial x}\right]_{i,j}^{n}+\left[\frac{\partial hCv^{E}}{\partial y}\right]_{i,j}^{n}+ \\ +\left[\frac{\partial}{\partial x}\left[D_h h\frac{\partial C}{\partial x}\right]\right]_{i,j}^{n}+\left[\frac{\partial}{\partial y}\left[D_h h\frac{\partial C}{\partial y}\right]\right]_{i,j}^{n}=\left[\frac{hC_{eq}-hC}{T_s}\right]_{i,j}^{n} \tag{3.61}$$

The bed-update is discussed next. Based on the gradients in the sediment transport the bed level changes according to:

$$(1-p)\frac{\partial z_b}{\partial t}+\frac{\partial S_x}{\partial x}+\frac{\partial S_y}{\partial y}=0 \tag{3.62}$$

where *p is the porosity and* S_x and S_y represent the sediment transport rates in *x*- and *y*-direction respectively, given by:

$$S_{x,i,j}^{n}=\left[\frac{\partial hCu^{E}}{\partial x}\right]_{i,j}^{n}+\left[\frac{\partial}{\partial x}\left[D_h h\frac{\partial C}{\partial x}\right]\right]_{i,j}^{n} \tag{3.63}$$

and

$$S_{y,i,j}^{n}=\left[\frac{\partial hCv^{E}}{\partial y}\right]_{i,j}^{n}+\left[\frac{\partial}{\partial y}\left[D_h h\frac{\partial C}{\partial y}\right]\right]_{i,j}^{n} \tag{3.64}$$

The bed-update is then approximated by:

$$\frac{z_{b,i,j}^{n+1}-z_{b,i,j}^{n}}{\Delta t}+\frac{f_{mor}}{(1-p)}\left[\frac{S_{x,i,j}^{n}-S_{x,i-1,j}^{n}}{\Delta x}+\frac{S_{y,i,j}^{n}-S_{y,i,j-1}^{n}}{\Delta y}\right]=0 \tag{3.65}$$

where f_{mor} represents a morphological factor to speed up the bed evolution (see e.g. Roelvink, 2006).

Transport formulations

The equilibrium sediment concentration can be calculated with various sediment transport formulae. At the moment the sediment transport formulation of Soulsby-van Rijn (Soulsby, 1997) has been implemented. The C_{eq} is then given by :

$$C_{eq}=\frac{A_{sb}+A_{ss}}{h}\left(\left(|u^{E}|^{2}+0.018\frac{u_{rms}^{2}}{C_d}\right)^{0.5}-u_{cr}\right)^{2.4}(1-\alpha_b m) \tag{3.66}$$

where sediment is stirred by the Eulerian mean and infragravity velocity in combination with the near bed short wave orbital velocity obtained from the wave-group varying wave energy. The combined mean/infragravity and orbital velocity have to exceed a threshold value, u_{cr}, before sediment is set in motion. The drag coefficient, C_d, is due to flow velocity only (ignoring short wave effects). To account for bed-slope effects on the equilibrium

sediment concentration a bed-slope correction factor is introduced, where the bed-slope is denoted by m and α_b represents a calibration factor. The bed load coefficients A_{sb} and the suspended load coefficient A_{ss} are functions of the sediment grain size, relative density of the sediment and the local water depth (see Soulsby [1997] for details).

3.7 Bottom updating

Avalanching

To account for the slumping of sandy material during storm-induced dune erosion avalanching is introduced to update the bed-evolution. Avalanching is introduced when a critical bed-slope is exceeded:

$$\left|\frac{\partial z_b}{\partial x}\right| > m_{cr} \tag{3.67}$$

Where the estimated bed slope is given by:

$$\frac{\partial z_b}{\partial x} = \frac{z_{b,i+1,j} - z_{b,i,j}}{\Delta x} \tag{3.68}$$

The bed-change within one time step is then given by:

$$\begin{aligned} \Delta z_b &= \min\left(\left(\left|\frac{\partial z_b}{\partial x}\right| - m_{cr}\right)\Delta x,\ 0.005\right) \quad , \frac{\partial z_b}{\partial x} > 0 \\ \Delta z_b &= \max\left(-\left(\left|\frac{\partial z_b}{\partial x}\right| - m_{cr}\right)\Delta x, -0.005\right) \quad , \frac{\partial z_b}{\partial x} < 0 \end{aligned} \tag{3.69}$$

where a threshold of 0.005 m has been introduced to prevent the generation of large shockwaves. The corresponding bed update is given by:

$$\begin{aligned} z_{b,i,j}^{n+1} &= z_{b,i,j}^{n} + \Delta z_{b,i,j} \\ z_{b,i+1,j}^{n+1} &= z_{b,i+1,j}^{n} - \Delta z_{b,i,j} \end{aligned} \tag{3.70}$$

To account for continuity, e.g. when sand is deposited within the wet part of the domain, the water level is also updated:

$$\begin{aligned} z_{s,i,j}^{n+1} &= z_{s,i,j}^{n} + \Delta z_{b,i,j} \\ z_{s,i+1,j}^{n+1} &= z_{s,i+1,j}^{n} - \Delta z_{b,i,j} \end{aligned} \tag{3.71}$$

Similar expressions are used for the subsequent avalanching in the y-direction.

3.8 Boundary conditions

3.8.1 Offshore flow boundary conditions

The offshore boundary is an artificial boundary which has no physical meaning. On the offshore boundary wave and flow conditions are imposed. In the domain waves and currents will be generated which need to pass through the offshore boundary to the deep sea with minimal reflection. One way to do this is to impose a weakly reflective-type boundary condition.

We chose to implement the formulation by Van Dongeren and Svendsen (1997) which in turn is based on Verboom et al. (1981) and is based on the Method of Characteristics. The boundary condition is implemented in the flow_bc subroutine.

The boundary conditions satisfy the following two necessary conditions:

1. the region outside the computation domain can influence the motion within the domain only through the incident (long) waves and through the currents along the boundaries; and
2. the (long) waves propagating out of the computational domain must be allowed to freely propagate through the open-ocean offshore boundary with minimal reflection.

By placing the open boundaries carefully, one can achieve weak local forcing near these boundaries. In practice this means that the offshore boundary is placed in sufficiently deep water, i.e. outside the shoaling zone. Then the dominant terms in the continuity and momentum equations near these boundaries are the nonlinear shallow water equations (3.31, 3.32, and 3.33).

For the general case of an arbitrary angle υ between the boundary at a point and the coordinate axes, one can follow the work of Abbott (1979) and Verboom *et al.* (1981) to derive the governing equations, which are valid for an arbitrary angle υ between the coordinate axes and the model boundary (Figure 3.3.a).

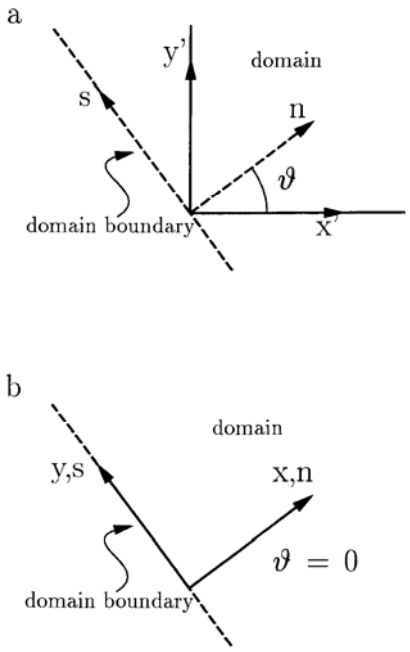

Figure 3-3 Coordinate system (a) for arbitrary angle υ between domain boundary and x-axis; (b) for υ=0

The derivation becomes simplified if the coordinate system is defined in a way the the x-axis is normally inward to the seaward boundary of the rectangular domain, which sets υ=0 (Figure 3.3.b). The governing equations derived following Abbott (1979) and Verboom *et al.* (1981) then simplify to

$$\frac{\partial \beta^-}{\partial t} = -(u-c)\frac{\partial \beta^-}{\partial x} - v\frac{\partial \beta^-}{\partial y} + c\frac{\partial v}{\partial y} + g\frac{\partial h_0}{\partial x} + F_{\beta^-} \tag{3.72}$$

$$\frac{\partial \beta^+}{\partial t} = -(u-c)\frac{\partial \beta^+}{\partial x} - v\frac{\partial \beta^+}{\partial y} - c\frac{\partial v}{\partial y} + g\frac{\partial h_0}{\partial x} + F_{\beta^+} \tag{3.73}$$

$$\frac{\partial \gamma}{\partial t} = -u\frac{\partial \gamma}{\partial x} - v\frac{\partial \gamma}{\partial y} - g\frac{\partial \eta}{\partial y} + F_{\gamma} \tag{3.74}$$

where, F includes all local forcing and friction terms for the motion, c is the wave celerity, and h_0 is the still water depth. The Riemann variable β^- is defined as

$$\beta^- = u - 2c = u - 2\sqrt{g(h_0+\eta)} \tag{3.75}$$

Here $\bar{u}$ is the depth-averaged velocity. The Riemann variable β^+ is similarly defined as $\beta^+ = u + 2c$. The γ-equation is the y-momentum equation, which has the Riemann variable

$$\gamma = v \tag{3.76}$$

The definition sketch in Figure 3.4 shows that β^- propagates in the negative *x*-direction, β^+ propogates in the positive *x*-direction, and γ in the *y*-direction. The forcing terms, *F*, in equations 3.72, 3.73, and 3.74 originate from the right-hand side of equations 3.31, 3.32, and 3.33, which imply that β^- , β^+, and γ are variables rather than constants.

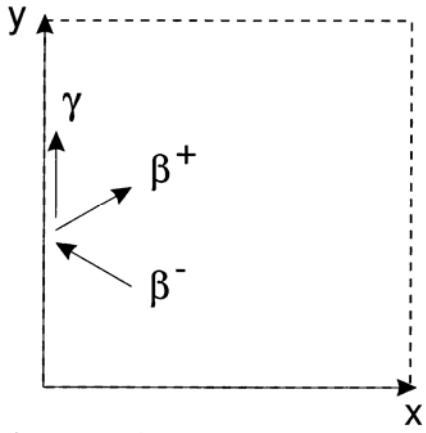

Figure 3-4 Definition sketch of the characteristics.

The offshore boundary conditions uses the outgoing β^- variant which contains information about the waves leaving the domain and the γ variant which propagates along the boundary. The latter is extra information which we will use to estimate the direction of the outgoing wave which is the innovation in Van Dongeren and Svendsen (1997).

The procedure is as follows: during the computation, at time step *n*, we know the values of η^n and the total velocities (u^n, v^n) at all points in the domain. The incoming wave is specified along the open boundaries through the *x* and *y* components of the particle velocities of the incident wave (u_{in}, v_{in}). The numerical integration of nonlinear shallow water equations will provide the values of the total η, *u*, and *v* for the interior points in the domain at time step *n+1*, and then the equivalent total (incoming plus outgoing) values need to be determined along the boundaries at *n+1*. In other words, given the incoming wave, the outgoing wave needs to be determined.

In XBeach we implement the lowest-order derived equations for the weakly reflective boundary conditions, with x=0 at the boundary. The outgoing wave angle (θ_r) and velocity in the *x*-direction (u_r) are solved for iteratively. For specifics on this derivation we refer to Van Dongeren and Svendsen (1997), the shorter outline is given below

The β^- is updated along the boundary only through (3.72) which discretized in Xbeach (similarly as the x-momentum equation) reads

$$\frac{\beta_{i,j}^{n+1} - \beta_{i,j}^{n}}{\Delta t} = -\left(u - c\right)\frac{\partial \beta^{n}}{\partial x}_{i,j} - v\frac{\partial \beta^{n}}{\partial y}_{i,j} + c\frac{\partial v^{n}}{\partial y}_{i,j} + g\frac{h_{\mathrm{i+1,j}}^{n} - h_{\mathrm{i,j}}^{n}}{x_{\mathrm{i+1,j}} - x_{\mathrm{i,j}}} + \frac{F_{x,i,j}}{\rho h_{u,i,j}} \qquad (3.77)$$

and is thus known at the time level *n+1*. We can then solve for the outgoing velocity u_r by expanding the Riemann variant (3.75) to lowest order as

$$\beta^- = u - 2\sqrt{g(h_0+\eta)} = u - 2\sqrt{gh_0}\left(1+\frac{1}{2}\frac{\eta}{h_0}\right) \tag{3.78}$$

We further have the identities

$$\begin{aligned} u &= u_i + u_r \\ \eta &= \eta_i + \eta_r \\ u_i &= \sqrt{gh_0}\,\eta_i \cos\theta_i \\ u_r &= -\sqrt{gh_0}\,\eta_r \cos\theta_r \end{aligned} \tag{3.79}$$

where the last two identities assume a wave propagating in shallow water with constant form where θ_i and θ_r are the angles of the incoming (known) wave and the outgoing (yet unknown) wave, relative to the x=0 boundary. Inserting these identities into (3.78) and re-arranging gives

$$u_r = \left(\frac{\cos\theta_r}{\cos\theta_r+1}\right)\left[\beta^- + 2\sqrt{gh_0} - u_i\left(\frac{\cos\theta_i-1}{\cos\theta_i}\right)\right] \tag{3.80}$$

All the terms on the right hand side are known except for θ_r which can be solved from the $\gamma = v$ variant as

$$\theta_r = \arctan\left(\frac{u_r}{v_r}\right) = \arctan\left(\frac{u_r}{v-v_i}\right) \tag{3.81}$$

Eqs. (3.80) and (3.81) are then solved iteratively to yield both u_r and θ_r. The final boundary condition is then the total velocity $u = u_i + u_r$ at the boundary at the time level *n+1*.

3.8.2 Offshore wave boundary conditions

At present, three options have been implemented to prescribe wave boundary conditions at the offshore boundary:

- Stationary wave boundary conditions; in this case a uniform, constant wave energy distribution is set, based on given values of Hrms, Tm01, direction and power of directional distribution function.

$$\begin{aligned} e_0(\vartheta) &= E_{mean}\frac{\cos^m\left(\vartheta-\vartheta_m\right)}{\sum_{\vartheta_{\min}}^{\vartheta_{\max}}\cos^m\left(\vartheta-\vartheta_m\right)\Delta\vartheta} \quad , \left|\vartheta-\vartheta_m\right| < \pi/2 \\ E_{mean} &= \frac{1}{8}\rho g H_{rms}^2 \end{aligned} \tag{3.82}$$

- Wave energy varying periodically in time (regular wave groups, bichromatic waves):

$$\begin{aligned} e_0(\vartheta) &= E_{mean}\frac{\cos^m\left(\vartheta-\vartheta_m\right)}{\sum_{\vartheta_{\min}}^{\vartheta_{\max}}\cos^m\left(\vartheta-\vartheta_m\right)\Delta\vartheta} * \frac{1}{2}\left(1+\cos\left(2\pi\left(\frac{t}{T_{long}}-\frac{y}{L_{long}}\right)\right)\right), \\ L_{long} &= \frac{c_g T_{long}}{\sin(\vartheta_m)} \end{aligned} \tag{3.83}$$

- Long-crested, irregular wave groups, where E is read in as a function of time; the timeseries is shifted along the y-axis to account for the oblique incidence.

$$e_0(\vartheta, y) = E_{t-\tau(y)} \frac{\cos^m(\vartheta - \vartheta_m)}{\sum_{\vartheta_{min}}^{\vartheta_{max}} \cos^m(\vartheta - \vartheta_m)\Delta\vartheta}, \qquad (3.84)$$

$$\tau(y) = \frac{y \sin(\vartheta_m)}{c_g}$$

3.8.3 Lateral flow boundary conditions

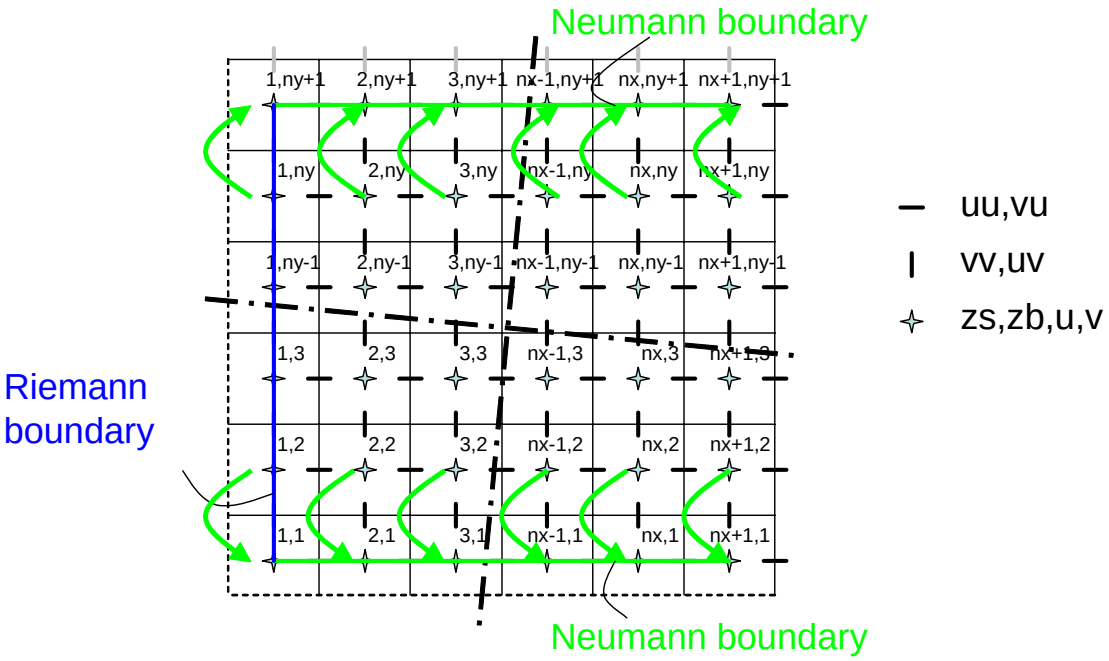

Figure 3-5: Stencil for Neumann-type boundary conditions.

For the lateral boundaries so-called Neumann boundaries are used, where the longshore water level gradient is prescribed, in this case set to zero. This type of boundary conditions has been shown to work quite well with (quasi-)stationary situations, where the coast can be assumed to be uniform alongshore outside the model domain. So far we have found that also in case of obliquely incident wave groups this kind of boundary conditions appears to give reasonable results, though rigorous testing still has to be done. The implementation consists of copying water levels from row 2 to row 1 and from row *ny* to row *ny+1*, and doing the same for the cross-shore (along-boundary) velocities. The alongshore velocities can now be computed from row 1 through row *ny*; no additional boundary conditions are required for the alongshore velocity.

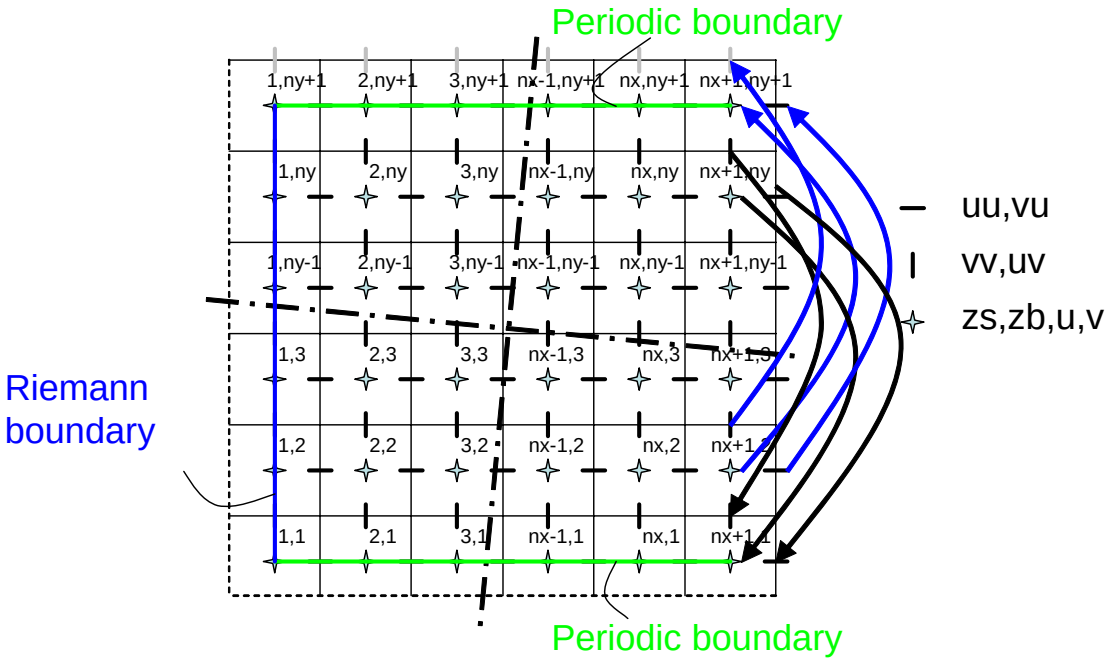

Figure 3-6: Stencil for periodic boundary conditions.

For the case of stationary or periodic variation of wave energy, alongshore periodic boundary conditions can be applied, where zs, uu and vv are simply copied from row 2 to row *ny+1* (blue arrows) **and** from row *ny* to row 1 (black arrows). This is only valid if the forcing is stationary or alongshore periodic at exactly the distance between row 1 and row ny, which amounts to a distance of (ny-1)*dy.

3.8.4 Lateral wave boundary conditions

For the lateral boundary conditions we make the following reasonable assumptions for the *incoming* wave energy:

- In the stationary case, we assume that the alongshore gradient of the wave energy is zero; this means we copy the value of one row inside the domain to the boundary, for the directional bins where the direction is *into* the model domain;
- In the instationary case, we assume that the gradient along the *crest* of the wave group is zero. The direction of the crest is derived from the local mean wave direction and the values at the boundary are determined by interpolation between the two points on the row inside around a virtual point taken along rhe crest direction; in the figure, for example, the value at point (3,1) is interpolated from points (2,2) and (3,2).

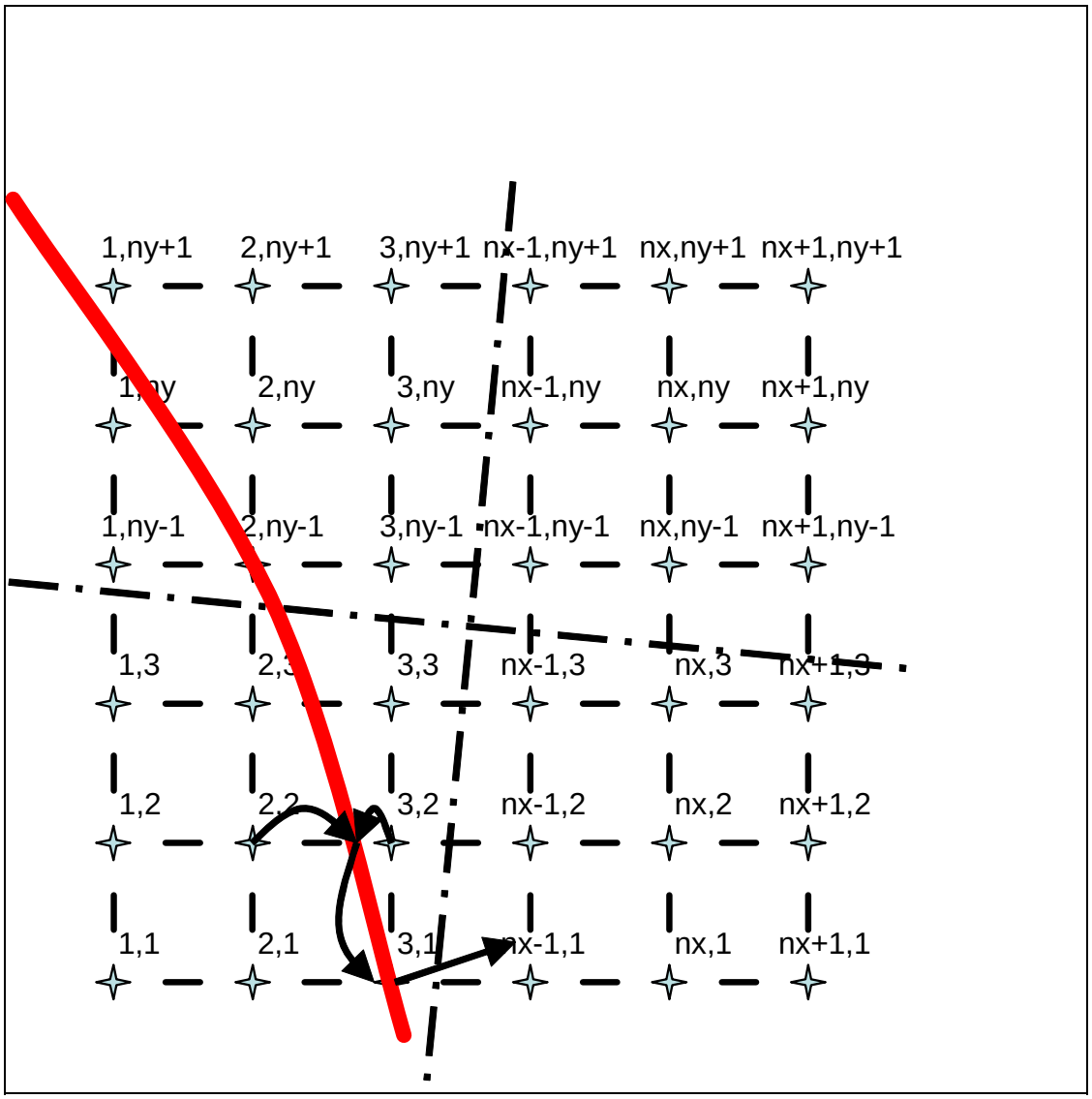
1,ny+1
2,ny+1
3,ny+1
nx-1,ny+1
nx,ny+1
nx+1,ny+1
1,ny
2,ny
3,ny
nx-1,ny
nx,ny
nx+1,ny
1,ny-1
2,ny-1
3,ny-1
nx-1,ny-1
nx,ny-1
nx+1,ny-1
1,3
2,3
3,3
nx-1,3
nx,3
nx+1,3
1,2
2,2
3,2
nx-1,2
nx,2
nx+1,2
1,1
2,1
3,1
nx-1,1
nx,1
nx+1,1

Figure 3-7:

4 Validation studies

4.1 Long wave propagation and numerical damping

The purpose of the first test is to check if the scheme is not too dissipative and that it does not create large errors in propagation speed.

A long wave with a small amplitude of 0.01 m and period of 80 s was sent into a domain of 5 m depth, grid size of 5 m and a length of 1 km. At the end, a fully reflecting wall is imposed. The wave length in this case should be sqrt(9.81*5)*80 = 560 m. The velocity amplitude should be sqrt(g/h)*amp = sqrt(9.81/5)*0.01 = 0.014 m. After the wave has reached the wall, a standing wave with double amplitude should be created.

As Figures 5.1 and 5.2 show, the model accurately represents this situation. There is hardly any dissipation, the wave length is very close to what it should be and there is no re-reflection off the seaward boundary.

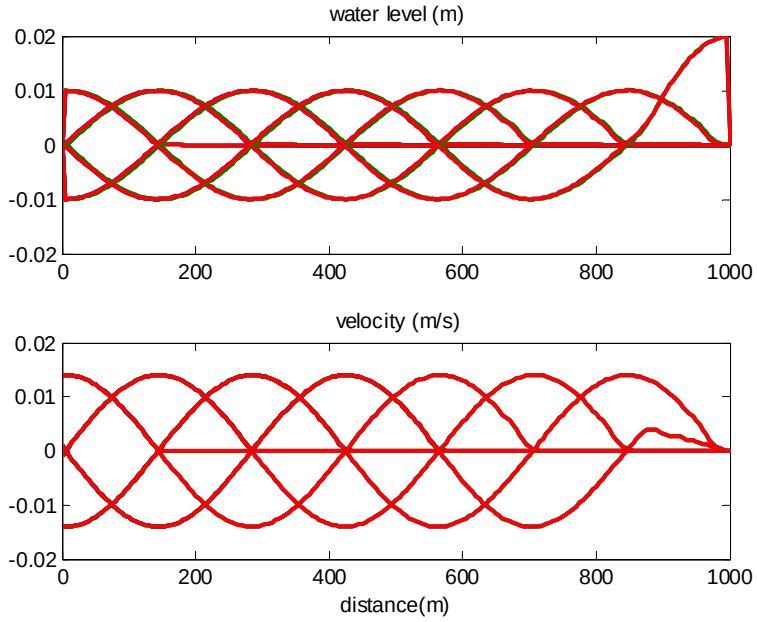

Figure 4-1 Snapshots of water level and velocity at T/4 intervals; just as wave hits wall.

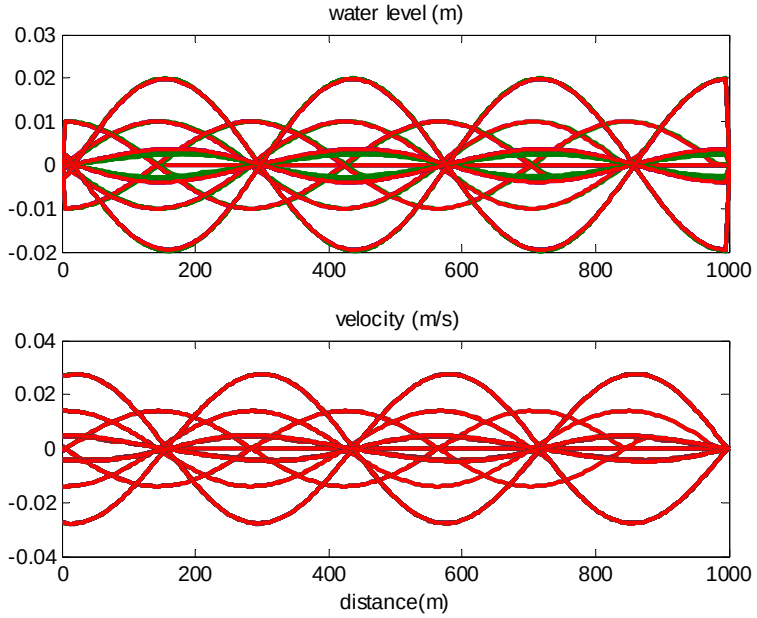

Figure 4-2 Same as 5.1, after long time.

4.2 Long wave runup on sloping beach; comparison with Carrier and Greenspan (1958)

The purpose of this test is to check the ability of the model to represent runup and rundown of long waves. To this end, a comparison was made with the analytical solution by Carrier and Greenspan (1958), which describes the motion of harmonic, non-breaking long waves on a plane sloping beach without friction.

In Figure 5.3 the model results for waves at an amplitude of ½ the breaking wave amplitude are shown, at 1/20 T intervals. The agreement is quite good, though there is very small disturbance/lag during rundown. Typical of the solution is that the profiles during runup and rundown should be exactly equal; apart from a small area near the water line during rundown, this is the case.

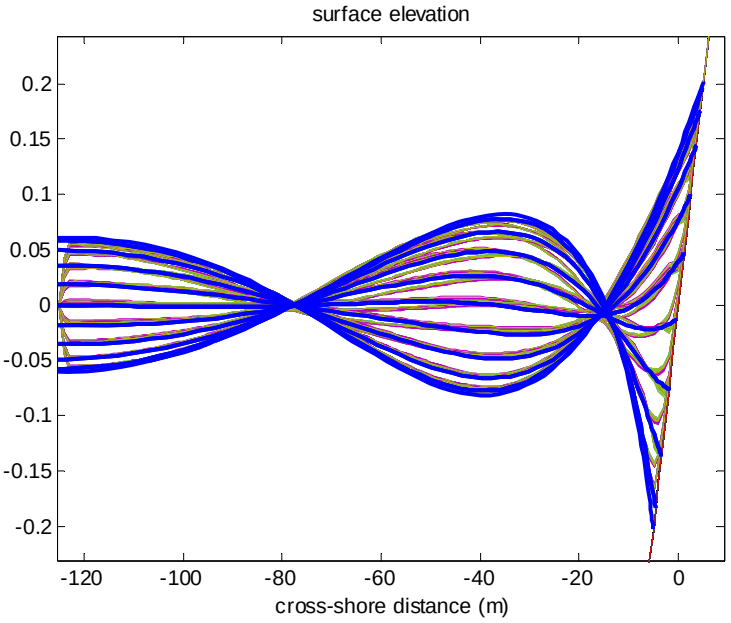

Figure 4-3 Surface elevation snapshots at 1/20 T intervals; model (thin brown lines) and analytical solution (thick blue lines).

4.3 Stationary wave propagation. dissipation and setup

The purpose of this test was to check the wave energy and momentum balance in stationary mode.

The Delta Flume test of Arcilla et al. (1993), test 2E was used. This test was carried out with an increased water level and significant dune erosion occurred during the test. Besides, extensive hydrodynamic, sediment transport and morphological measurements were carried out.

The dissipation model of Baldock (1998) was applied, with a gamma value of 0.8. The results show a good agreement for the Hrms wave height, mean setup and rms value of the orbital velocity.

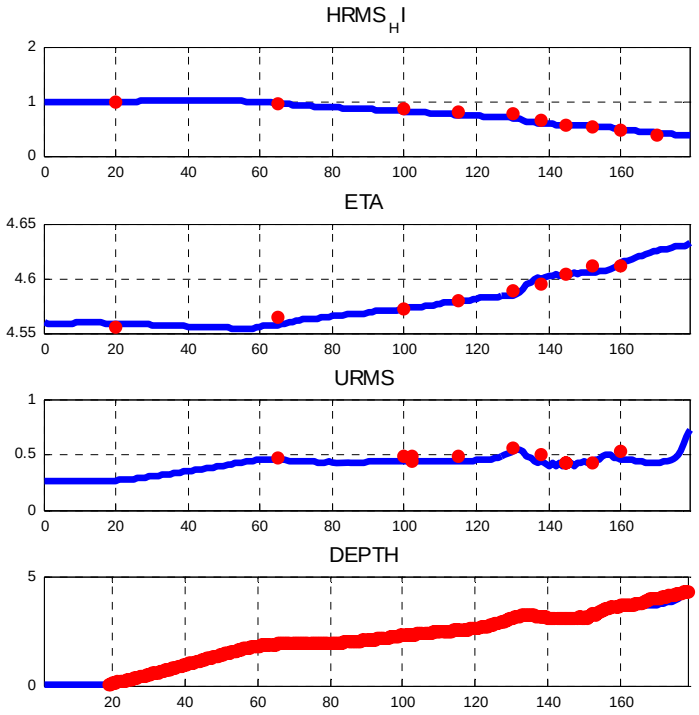

Figure 4-4 Wave height decay, setup and orbital velocity, modelled in stationary mode (drawn blue line) vs. measured (red dots). LIP 11D test 2B.

4.4 Nonstationary surf zone flows in large-scale flume test

The purpose of this test was to verify the hydrodynamics of the model when run in nonstationary mode, viz. with a time-varying wave energy imposed at the offshore boundary.

The same test case as before was used. The time series of wave energy was generated by a procedure described in Roelvink (1993b), based on a JONSWAP spectral shape as was applied in the test. Zero-order steering was applied in the flume test; therefore no incident bound long wave was imposed in the numerical experiment.

The results are given in Figures 5.5 and 5.6, and show that the short wave decay and the generated long waves are represented quite well, both in surface elevation and in near-bottom velocity. Also the time-averaged water level is represented quite accurately.

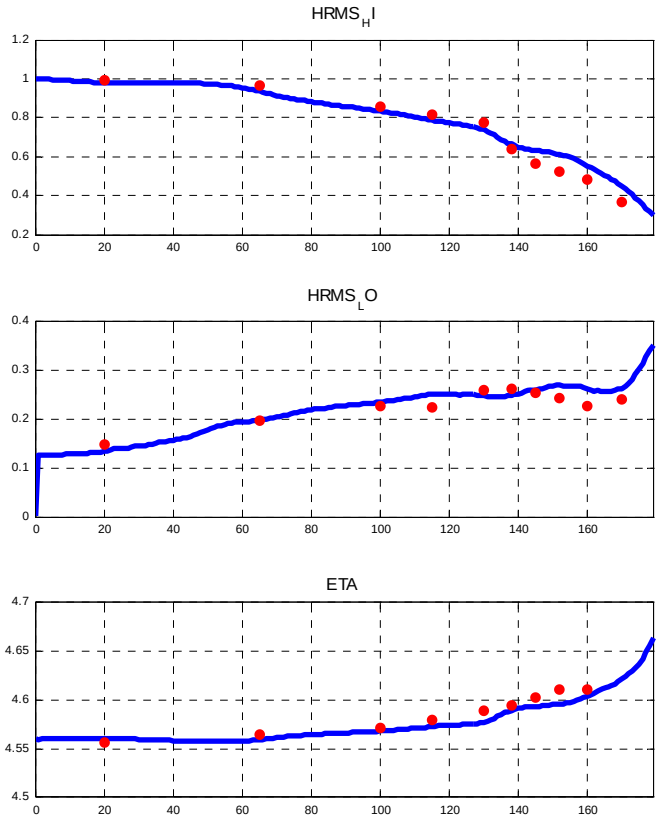

Figure 4-5 Wave height, LF wave height and setup, nonstationary model results vs measurements from LIP 11D, test 2E.

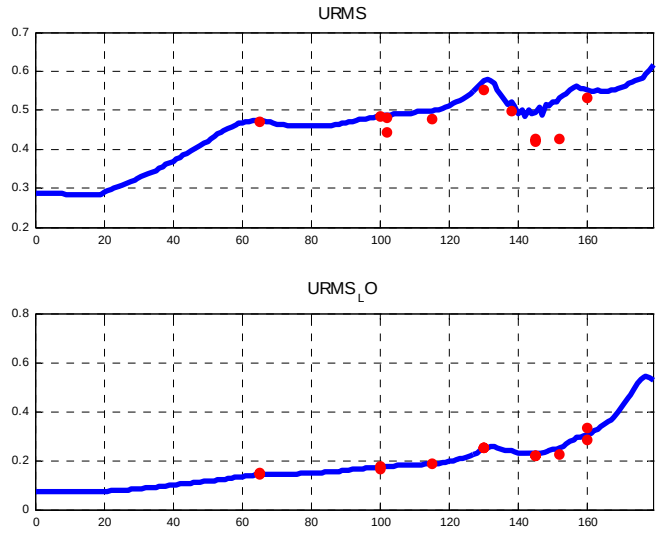

Figure 4-6 Orbital velociity and long wave rms velocity , nonstationary model results vs LIP 11D test 2E.

4.5 Absorbing-generating boundary condition tests

To test the implementation of the weakly reflective boundary conditions to the offshore open boundary, a simulation was performed using a long-shore and cross-shore uniform

bathymetry, with a wall near the shoreward boundary. The uniform water depth was 5 meters. The domain consisted of 141 grid cells in the cross-shore and 51 grid cells in the longshore. The cross-shore and longshore spacing were 5m and 20m respectively. The incident wave angles were 0° and 30° off normal (270° and 240° Nautical), with an offshore long wave height of 0.05 m and a period of 70s. The simulation ran for 1000 timesteps.

The implementation of the weakly reflective boudary conditions to the offshore boundary improved the water surface elevation prediction at the offshore boundary, as well as the stability of the simulation when forcing with oblique waves. Figure 4.7 illustrates these improvements to Xbeach when implementing the updated boundary conditions at multiple time steps for the case of normal incidence. The figure shows the new boundary conditions in solid red and the old boundary conditions in dashed blue. The differences are minimal except near the boundary as was expected. The virtue of the new boundary condition is for the oblique incident case.

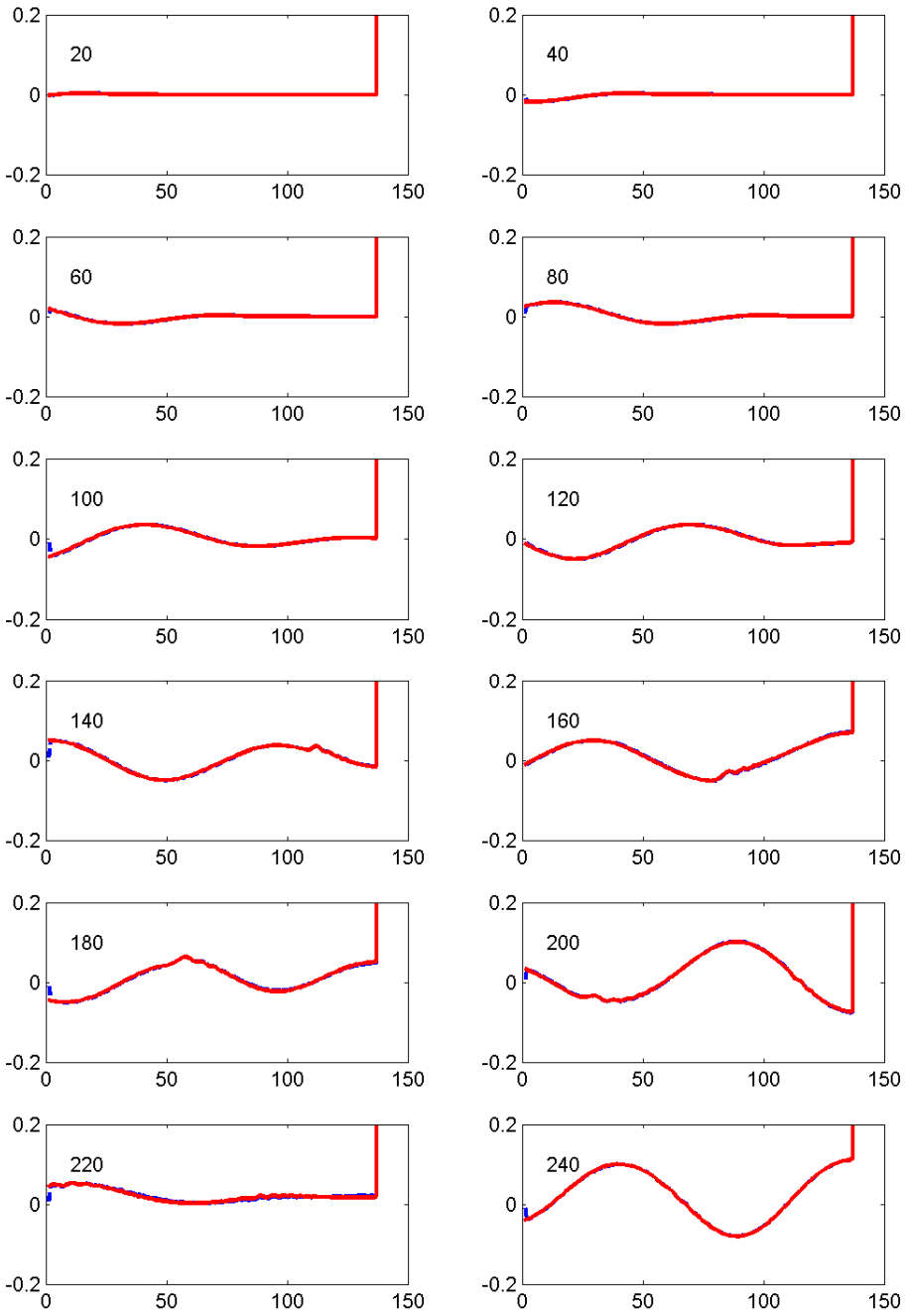

Figure 4-7 Water surface elevation comparisons between Xbeach simulations with new weakly reflected offshore boundary condition implemented (red solid-dotted line) and with old weakly reflective condition (blue dashed-dotted line) at multiple times in the simulation.

Figure 4.8 shows snapshots of the oblique-angle case. The simulation shows that from still water the incoming long wave is propagated towards the wall (on the right) and that the

reflections (preceded by a noisy spin-up front) from that wall set up a checkerboard pattern of a shortcrested wave which is propagating in the longshore direction. The reflections pass through the offshore boundary to the "outer" (unmodelled) domain with minimal reflections, which can be seen as the absence of any lingering noise in the computational domain. For this case we applied a longshore periodic boundary condition to prove the correct implementation of the offshore boundary condition.

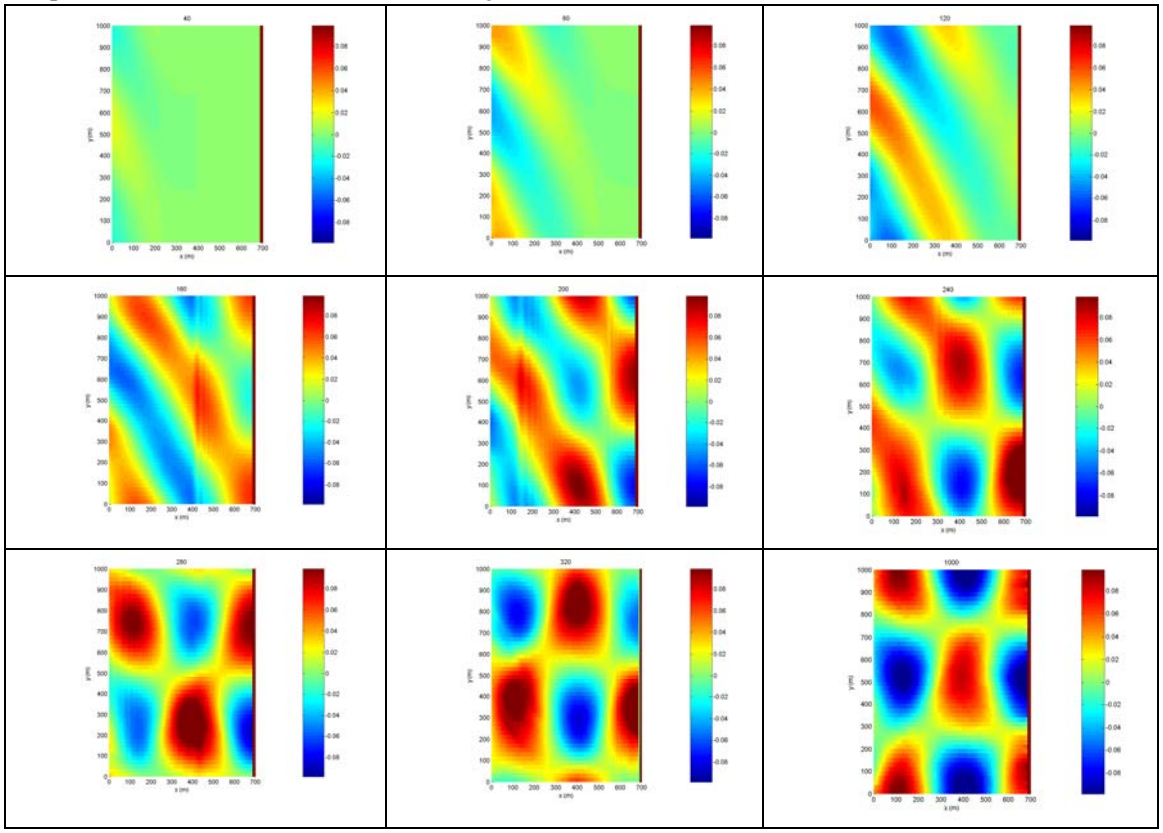

Figure 4-8 Water surface elevation for the case of an obliquely-incoming wave at a 30^{o} angle with the normal. Waves are incoming on the left boundary and propagate to the right wall, where they reflect. The short-crested pattern propagates to the top of the frames. The figures show 9 time instances showing the start-up of the incoming wave (top 3 panels), the reflected wave from the wall (middle three panels) and the resulting short-crested pattern (lower 3 panels).

The simulation is repeated for the same conditions, except a domain which has twice the alongshore length and has Neumann boundary conditions on the lateral boundaries instead of periodic boundaries. The figure shows a consistent picture compared to the previous case, except for the area at the bottom of the figure which now has a diffraction zone. The shortcrested wave is transmitted through the upper boundary correctly. This simulation shows that Neumann-type boundary conditions can be applied if care is taken that part of the domain near the inflow boundary is affected by diffraction effects. The computational domain must therefore be larger in the longshore extent than the domain of interest.

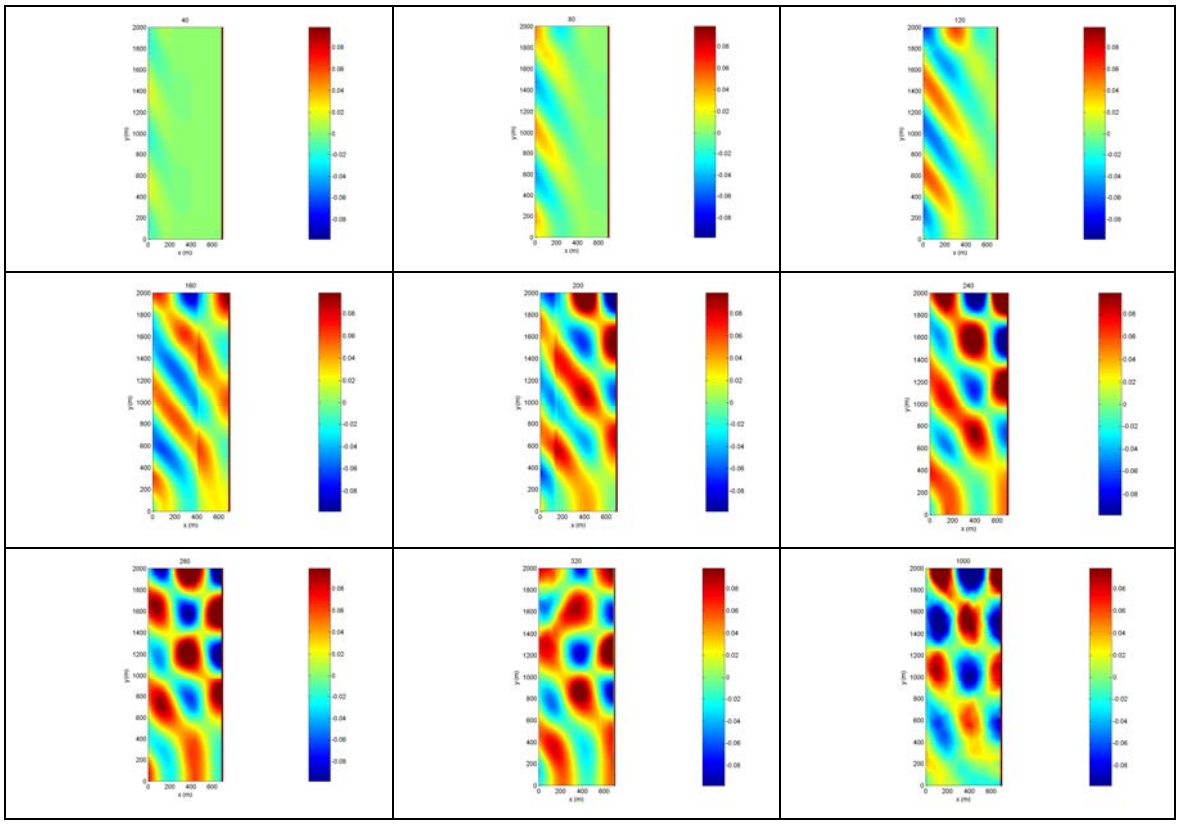

Figure 4-9 Water surface elevation for the case of an obliquely-incoming wave at a 30^{o} angle with the normal. Waves are incoming on the left boundary and propagate to the right wall, where they reflect. The short-crested pattern propagates to the top of the frames. The figures show 9 time instances showing the start-up of the incoming wave (top 3 panels), the reflected wave from the wall (middle three panels) and the resulting short-crested pattern (lower 3 panels). The domain has Neumann boundary conditions at the top and bottom boundaries.

4.6 Dune erosion in large-scale flume.

The purpose of this test was to verify the dune erosion modelling in nonstationary mode.

LIP (1993) Delta Flume Tests

The model was run for 0.8 hours of hydrodynamic time with a morphological factor of 10, effectively representing a morphological simulation time of 8 hours.

A key element in the modelling is the avalanching algorithm; although the surfbeat waves that are explicitly modelled run up and down the upper beach, without a mechanism to transport sand from dry to wet the dune erosion process will not happen. A relatively simple approach, whereby an underwater critical slope of 0.15 and a critical slope above water of 1.0 were applied, proves to be quite successful in representing the retreat of the upper beach and dune face. A grid resolution of 1 m was applied. In Figure 5.7 The measured and modelled bed evolution is shown, which looks quite promising in the upper region. The behaviour of the bar at approx. 135 m is not represented well; for this, additional processes

such as the effect of surface rollers and wave asymmetry/skewness have to be taken into account.

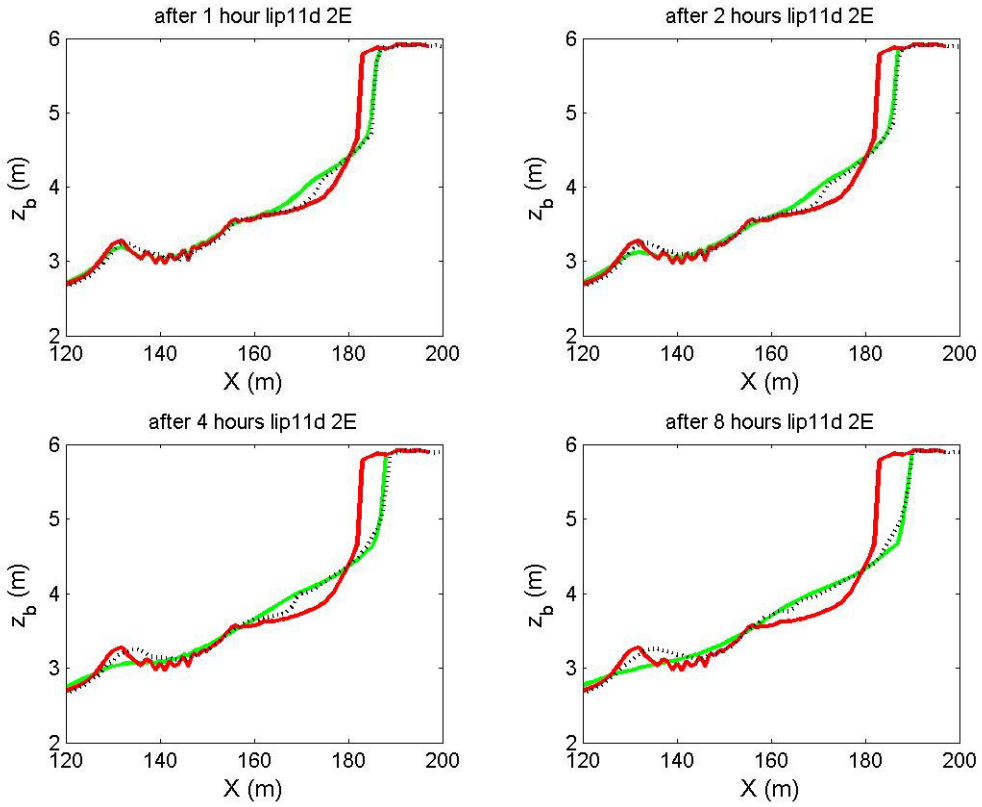

Figure 4-10 Measured and modelled bed level after 1, 2, 4 and 8 hours of wave action, for a water level of 4.56 m above the flume bottom.

Delta Flume 2005 experiments

In addition to the validation of the modelled dune erosion processes using the experimental data from the LIP (1993) Delta Flume Tests, the model was run using the same numerical settings for a Delft (2005) dune erosion test. We considered Test 3 with an H_{rms} = 1.06 m, T_{m02} = 6.20 s and a still water level of 4.5 m. The simulation was run for 0.6 hrs of hydrodynamic time with a morphological factor of 10, which results in a morphodynamic simulation time of 6 hrs. Using exactly the same input setting otherwise the figure shows that in 6 hrs the computed dune erosion profile is very similar to the final measured profile, except for the formation of the offshore bar. This physics of bar formation are not yet implemented (and strictly speaking irrelevant for dune erosion/overwash)

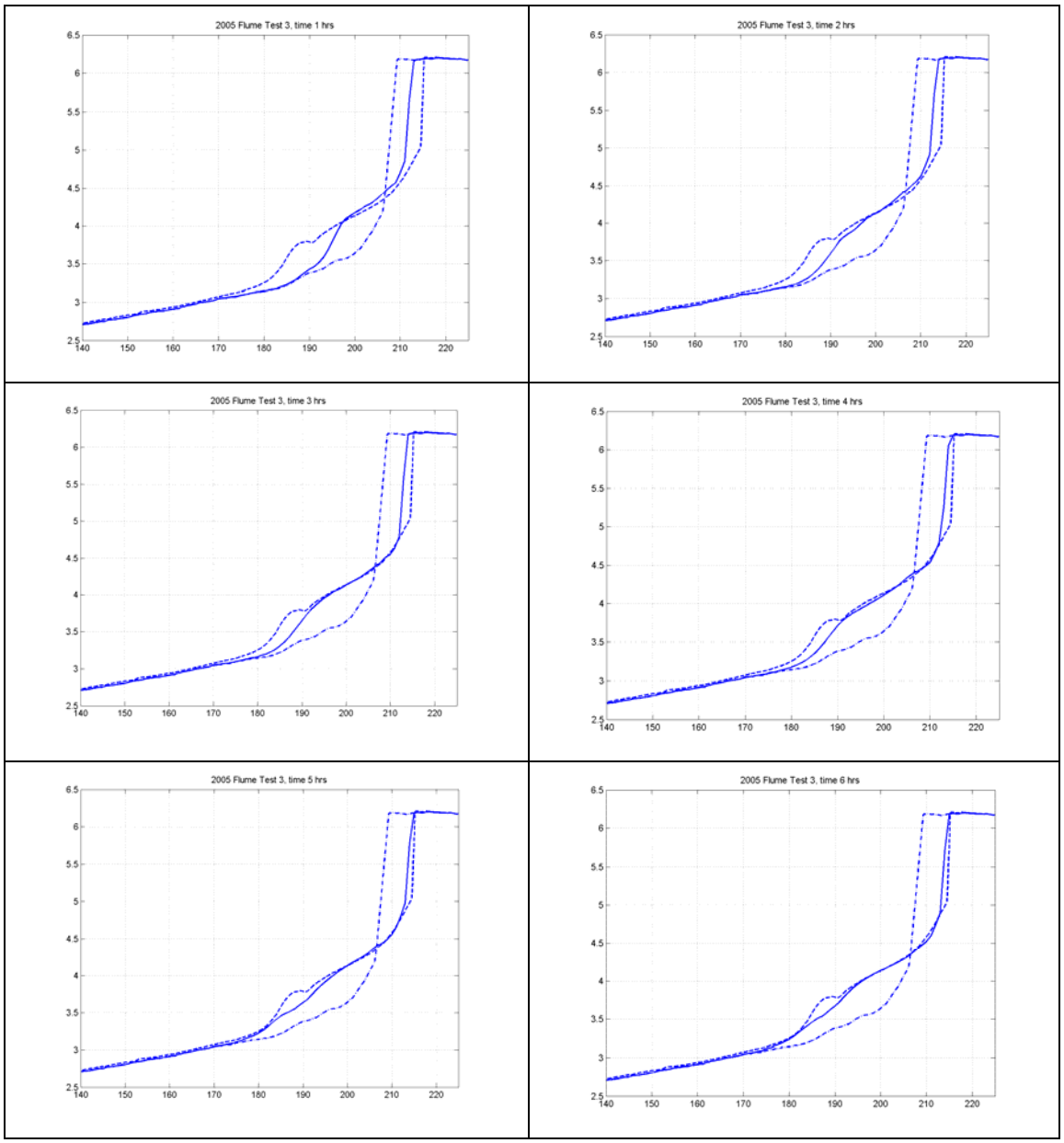

Figure 4-11: Dune erosion profiles: computed (solid line), initial (dash-dotted line) and final measured (dashed line)

4.7 Model formulation sensitivity studies

Five simulations are presented (Table 4.1) to examine model sensitivity to long waves and short waves. Simulation results are compared with test 2E from the Delta Flume experiment of Arcilla et al. (1993).

Test	Description	Realization
T1	Both long-and short waves	Default model settings
T2	Short waves only	$F_x = 0$ and no wave group varying wave energy at model boundary.
T3	Long waves only	$U_{rms} = 0$ and $U_S = 0$
T4	Long waves with short wave sediment stirring	$U_S = 0$
T5	Long waves, with short wave mass flux	$U_{rms} = 0$

Table 4.1: Overview simulations to examine model sensitivity to long -and short waves

Simulation results for default model settings (test T1) in which both long and short waves are present are shown in Figure 4-12. In shore ward direction short wave energy dissipates whereas long wave energy increases and exceeds simulated short wave energy. Close to the dune face strong return flows and large sediment concentrations are simulated. No measurements from LIP test 2E are available to validate simulated peak values. Simulated –and measured sediment transports and profile evolution compare well in the near shore area where the fore shore evolves due to dune erosion.

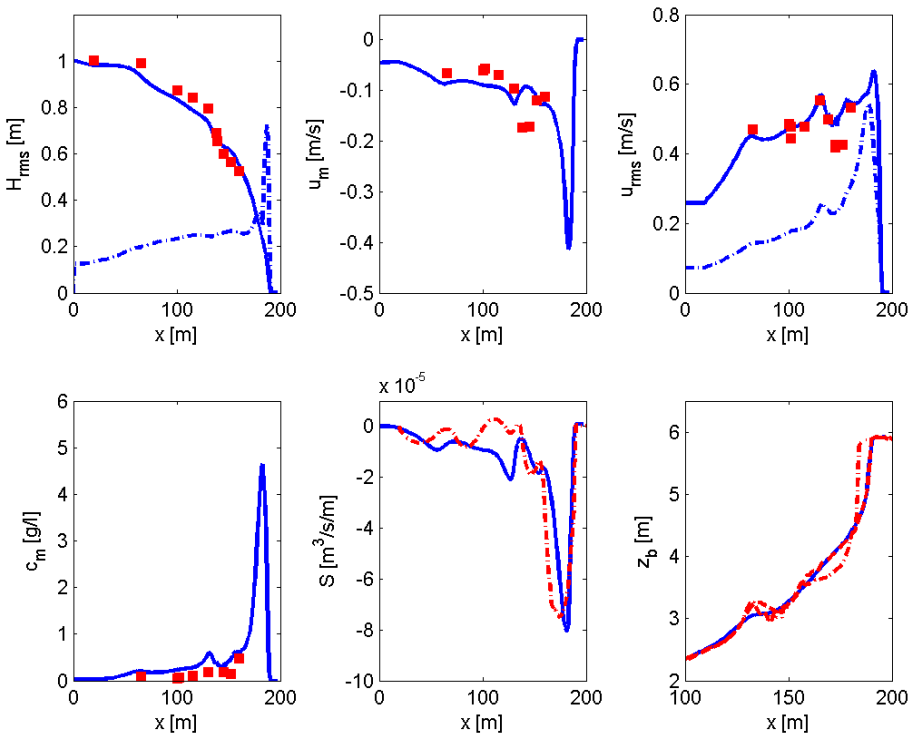

Figure 4-12: simulation results test T1, upper left panel short (solid) -and long (dashed dotted) wave height compared with measurements (squares), middle panel depth averaged return flow (solid) compared with measurements (squares), upper right panel short (solid) –and long (dashed dotted) wave orbital motion compared with measurements (squares), lower left panel depth averaged sediment concentration (solid) compared with measurements (squares), lower middle panel sediment transports (solid) compared with measurements (dashed dotted) and lower right panel post test profile (solid) compared with measured initial (dashed dotted) -and post test profile (dashed).

Imposing only short waves (Wave forces in NSWE are set to zero (F_x = 0) and no wave group varying wave energy at model boundary) computed profile evolution due to dune erosion significantly deviates from test T1 and measurements (Figure 4-13). In front of the dune face a peak in undertow velocities -and sediment concentrations is observed however their magnitudes have reduced significantly. As a result sediment transports and dune erosion volumes lag behind those observed during the experiment. In addition simulated dune foot is positioned vertically lower in test T2 as during test T1. It seems long waves smooth the developing fore shore to a more equilibrium profile looking shape.

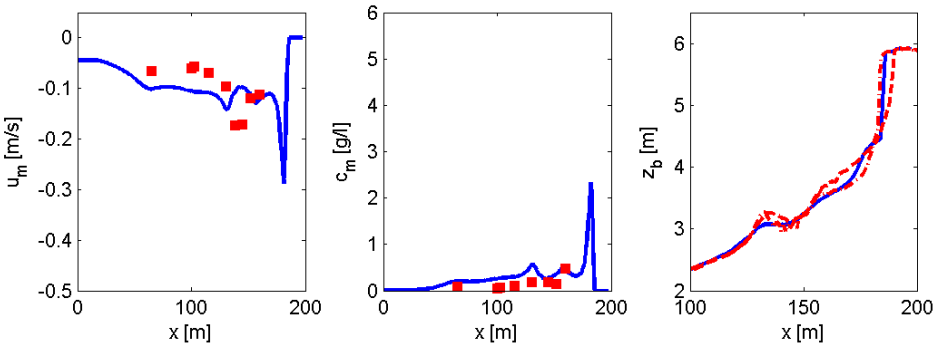

Figure 4-13: simulation results test T2, left panel depth averaged return flow (solid) compared with measurements (squares), middle panel depth averaged sediment concentration (solid) compared with measurements (squares), and right panel post test profile (solid) compared with measured initial (dashed dotted) -and post test profile (dashed).

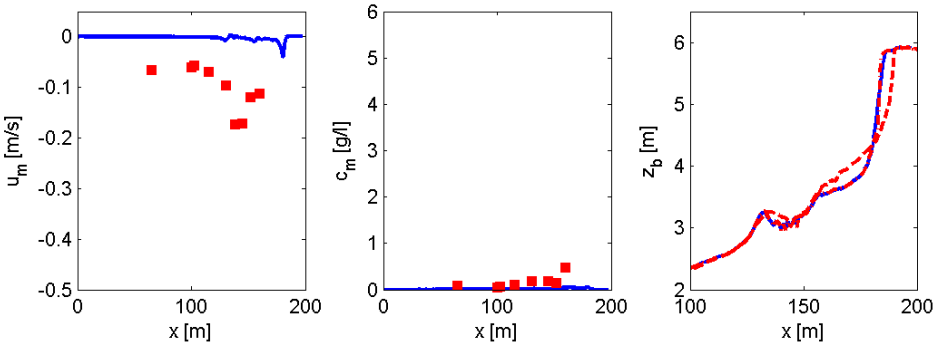

Figure 4-14 simulation results test T3, left panel depth averaged return flow (solid) compared with measurements (squares), middle panel depth averaged sediment concentration (solid) compared with measurements (squares), and right panel post test profile (solid) compared with measured initial (dashed dotted) -and post test profile (dashed).

Simulation results for only long waves are approximated by setting the Stokes drift U_S –and the short wave orbital motion U_{rms} to zero (Figure 4-14). As a result there is not a short wave driven undertow (Equation 3.39 and 3.40) and there is not sediment stirring due to short waves. Despite the observed influence of long waves in test T2 a model driven by only long waves is not capable to reproduce the profile evolution during LIP test 2E. Considering model results for test T2 and T3 it seems that interaction of long waves and short waves is important which will be further examined below.

Two additional simulations were produced to obtain better understanding in the driving hydrodynamics important to dune erosion within the model. In test T4 a simulation with long waves and short wave sediment stirring (and without short wave driven undertow) is

performed (Figure 4-15). The profile evolution is comparable to that of test T3 and does not agree with observations during LIP test 2E. Though simulated near shore sediment concentrations are large no undertow is present to advect sediment off-shore. Finally results for test T5 are presented in Figure 4-16. In this simulation with long waves and short wave driven undertow (and without short wave stirring) computed profile evolution compares much better with measurements.

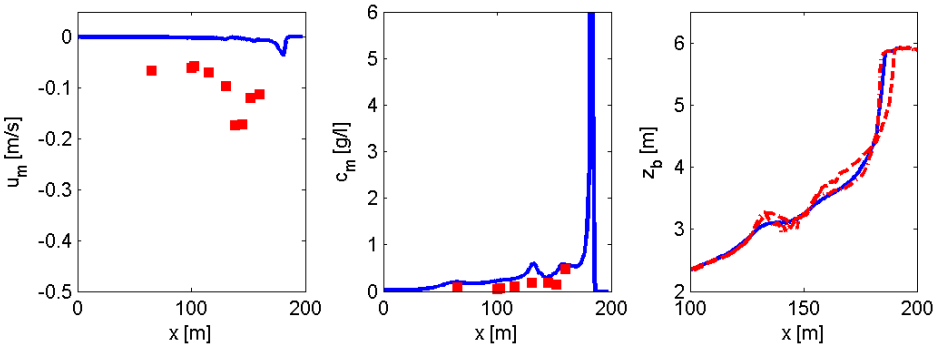

Figure 4-15 simulation results test T4, left panel depth averaged return flow (solid) compared with measurements (squares), middle panel depth averaged sediment concentration (solid) compared with measurements (squares), and right panel post test profile (solid) compared with measured initial (dashed dotted) -and post test profile (dashed).

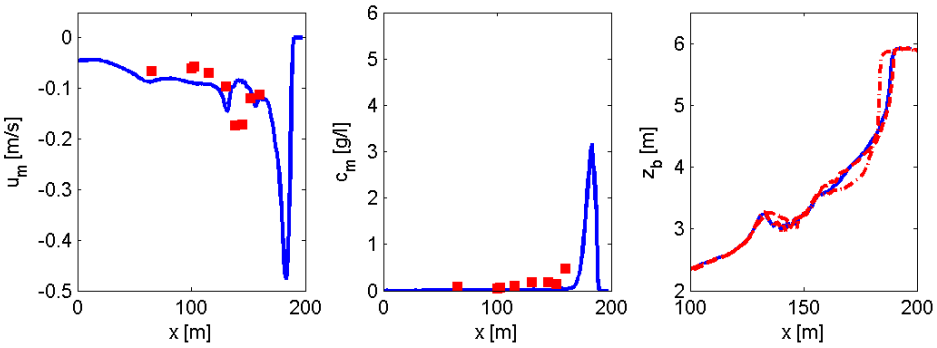

Figure 4-16 simulation results test T5, left panel depth averaged return flow (solid) compared with measurements (squares), middle panel depth averaged sediment concentration (solid) compared with measurements (squares), and right panel post test profile (solid) compared with measured initial (dashed dotted) -and post test profile (dashed).

To simulate dune erosion requires large sediment concentrations and a strong undertow in front of the dune face. Within the present model the undertow is mainly short wave driven whereas large near shore sediment concentrations are best explained by mean -and long wave flows. To validate simulated near shore sediment concentrations and undertow requires measurements in the inner surf and swash available from a large scale dune erosion tests recently conducted (Van Gent et al, 2006).

4.8 Dune erosion and overwash field tests

This work is based on the measurements and analysis of Assateague Island, Maryland, USA after large-scale morphological evolution was observed. The study is described by Jimenez, Sallenger, and Fauver (2006): Sediment Transport and Barrier Island Changes During Massive Overwash Event, presented at ICCE 2006, San Diego.

Jimenez *et al.* (2006) analyzes the response of sandy dunes on Assateague Island to extreme storm impacts. Two consecutive northeasters attacked the barrier island during late January and early February, 1998. The bathymetry was measured using Lidar in September 1997 and again February 9th and 10th, 1998 after the two storms had subsided.

Three types of dunes were identified by Jimenez *et al.* (2006) (Figure 4-17, the seaward side is to the right).

Profile A (upper left panel, initial profile is in black) was characterized by a steep faced dune, where the maximum run-up exceeded the dune crest height, and a mildly sloping dune back.

Profile type B is a double-peaked dune profile and has two different shapes. Profile B1 (upper right panel) is characterized by a primary and secondary dune, both of which are lower than the maximum run-up height and which are separated by a valley. Profile B2 (bottom left) has two peaks of which the seaward one is lower. The backside of the barrier of either type is therefore either characterized by a secondary dune line (profile B1) or a taller crest of the dune (profile B2) which prevents the eroded sand from being transported to the backside of the dune.

The height of the dune crest of profile C (lower right panel) exceeds the maximum run-up height, therefore little sediment is transport to the backside of the island, and mostly slumping occurs along the dune face.

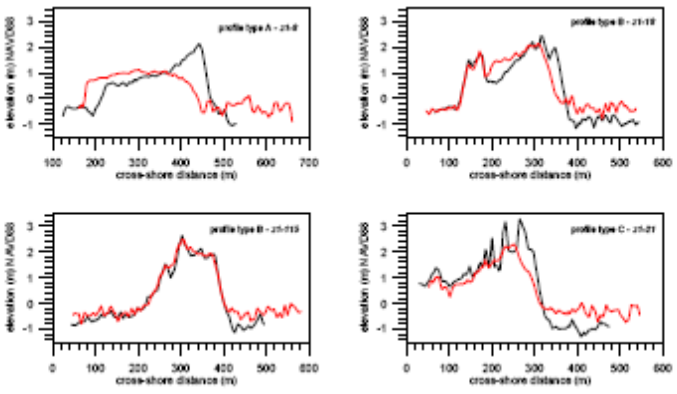

Figure 4-17 Jimenez *et al.* (2006) dune profiles denoting the three types identified (A, B, and C)

In this study, we implemented the measured profiles in a schematic way in order to simulate and validate the principal processes. The theoretical profiles we implemented in our Xbeach simulations are shown in Figure 4-18 (seaward side to the right).

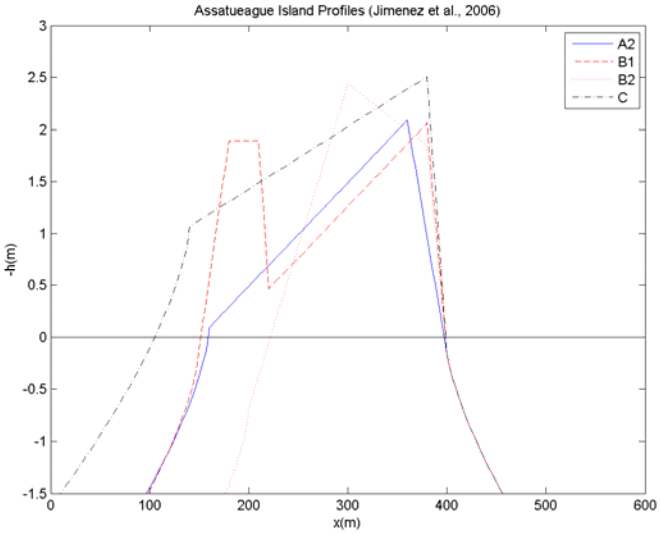

Figure 4-18 Xbeach Assateague Island profiles

The simulations were run for a 48 hour time periods using the following settings: morfac=5, z_{s0}=1.25(to mimic storm surge and similar to what was observed by Jimenez *et al.*(2006)) H_{rms0}=3m, T_{m01}=7.5s, T_{long}=60s, θ_{min}=-4.5º, θ_{max} =4.5º, $\Delta\theta$=2º, θ_0=0º, and a CFL=0.5. The offshore wave height and mean period with in the range of observations in this area as described by Jimenez *et al.*(2006) (CERC station MD002). The model domain is 1000m in the cross shore, with a Δx=2m, and 15m in the longshore, with a Δy=5m. The input bathymetry is longshore uniform. The weakly reflective boundary condition as described by Van Dongeren and Svendsen (1997) is applied to the offshore boundary.

The resulting four profiles are shown in Figure 4-19. The characteristic evolution described by Jimenez *et al.* (2006) is consistent with much of the profile evolution predicted by Xbeach.

Jimenez *et al.* (2006) observed that profile A became wider, flatter, with large quantities of eroded sediment deposited on the back side of the barrier island, due to the consistent wave over-topping. The model replicates this behavior, except for the setback of the entire profile which is seen in the measurements.

The evolution of profile B1 is noticeably different, in that the erosion, due to the wave overtopping, was not deposited between the primary and secondary dunes, as was observed by Jimenez *et al.* (2006). The model does replicate the dune face avalanching and nearshore deposit.

Profile B2 was observed to have little transport to the backside of the dune, and most transport occurring on the face of the seaward-side of the dune, resulting in slope reduction and general barrier narrowing in the model. Interestingly, while this dune erosion behavior would be expected in the field data, it shows very little morphological change, except for some nearshore deposits.

Jimenez *et al.* (2006) observed, in general, profile C to lower in height, the seaward dune slope to become smaller, and seaside retreat of the shoreline resulting in barrier narrowing. The model predicts much of the same behavior except that the intersection of the dune face profiles which in the data is at MSL 0 and in the model at MSL +1 (the surge level)

For Profiles A, B1, and C, the seaward face of the dunes were observed to retreat and the beach face slope decreased, which is consistent with the predictions made by Xbeach.

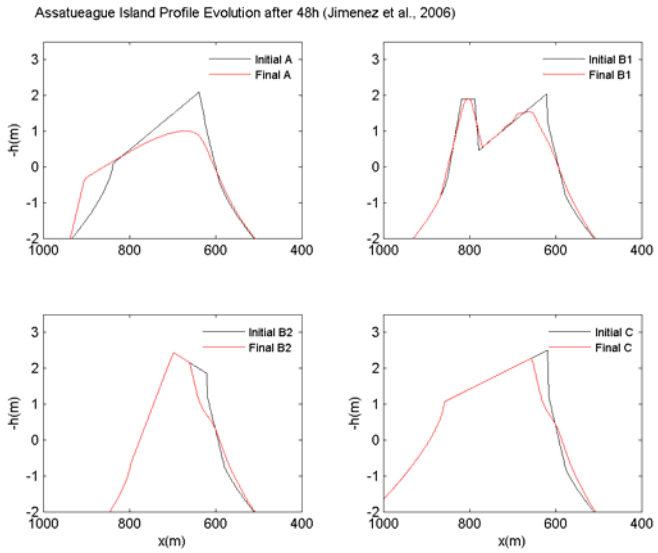

Figure 4-19 Resulting profile evolution after 48 hour simulations using Xbeach.

With the exception of the profile evolution of the dune valley of profile B1 and the dune face erosion of profile C, Xbeach performed well at simulating the general evolution patterns as observed by Jimenez *et al.* (2006).

5 Running the model

5.1 Input

The source code has been extensively tested in a version compiled under Compaq Visual Fortran version 6.6. We expect it to compile with only minor problems under other compilers and/or under Linux, since only standard Fortran 90/95 is used.
Once an executable has been created, it will be called xbeach.exe.

This executable will look for an input file named ‘params.txt’ in the case directory from which it is started. This input file contains the reference to a bathymetry file, which should contain, for each of ny+1 rows, nx+1 depth values, which may be defined positive downward or upward, depending on a keyword ‘posdwn’ that may be 1 (depth positive downward) or -1 (positive upward).

The ‘params.txt’ file contains grid and bathymetry info, wave input, flow input and morphological input. Table 5.1 below contains a description of the keywords, the default values and recommended minimum and maximum values.

Grid parameters

Keyword	Default	Minimum	Maximum	Unit	Description
nx	50	2	10000	[-]	number of grid cells x-direction
ny	2	2	10000	[-]	number of grid cells y-direction
dx	10.	.1	1000.	`[m]	grid size x-direction
dy	10.	.1	1000.	[m]	grid size y-direction
xori	0.	-1e9	1e9	[m]	x-origin of grid in world coordinates
yori	0.	-1e9	1e9	[m]	y-origin of grid in world coordinates
alfa	0.	-360.	360.	[deg]	angle of grid w.r.t. East

Wave input parameters

Keyword	Default	Minimum	Maximum	Unit	Description
Hrms	1.	0.	10.	[m]	Hrms wave height
Tm01	10.	1.	20.	[]	Tm01 wave period
dir0	30.	-90.	90.	[]	mean wave direction (Nautical convention)
m	10	2	128	[]	power in cos^m directional distribution
nt	2000	1	1000000	[]	max. number of time steps
hmin	0.01	0.001	1.	[m]	threshold water depth
gammax	5.	.4	5.	[-]	maximum ratio Hrms/hh
Tlong	80.	20.	300.	[s]	wave group period for case instat=1
gamma	0.6	0.4	.9	[-]	breaker parameter in Baldock or Roelvink formulation
alpha	1.0	0.5	2.0	[-]	wave dissipation coefficient
delta	0.0	0.0	1.0	[-]	fraction of wave height to add to depth in computation of celerity
n	5.0	5.0	20.0	[-]	power in roelvink dissipation model
rho	1025.0	1000.0	1040.0	[kg/m3]	water density
g	9.81	9.7	9.9	[m/s2]	acceleration of gravity
thetamin	-80.	-180.	180.	[deg]	lower directional limit (angle w.r.t computational x-axis)
thetamax	80.	-180.	180.	[deg]	upper directional limit (angle w.r.t computational x-axis))
dtheta	10.	0.1	20.	[deg]	directional resolution (deg)
wci	0	0	1	[-]	option wave/current interaction 0/1
break	2	1	2	[-]	option breaker model (1=roelvink, 2=baldock)
instat	1	0	2	[-]	option time-varying wave b.c. (0=stationary, 1=regular wave groups, 2=long-crested random wave groups)
roller	1	0	1	[-]	option to turn off/on roller model (0/1) (switch not implemented yet)
beta	0.15	0.05	0.3	[-]	breaker slope coefficient in roller model

Flow input parameters

Keyword	Default	Minimum	Maximum	Unit	Description
C	40.	20.	100.	[m^1/2/s]	Chezy value
eps	0.1	0.001	1.	[m]	threshold depth
umin	0.1	0.001	5.	[m/s]	threshold velocity upwind scheme
zs0	0.0	-5.	5.	[m]	initial water level
tstart	1.	0.	1000000.	[s]	start time of simulation
tint	1.	.01	100000.	[s]	time interval output
tstop	2000.	1.	1000000.	[s]	stop time simulation
CFL	0.2	0.1	0.9	[-]	maximum courant number
nuh	0.5	0.0	1.0	[m2/s]	horizontal viscosity coefficient

Morphology input parameters

Keyword	Default	Minimum	Maximum	Unit	Description
dico	1.	0.	10.	[m2/s]	diffusion coefficient
D50	0.0002	0.00005	0.001	[m]	D50 grain diameter
D90	0.0003	0.00005	0.001	[m]	D90 grain diameter
rhos	2650	2400.	2800.	[kg/m3]	sediment density
morfac	0.0	0.	1000.	[-]	morphological factor
morstart	300.	0.	100000.	[s]	start time morphological updating
wetslp	0.3	0.1	1.	[-]	critical avalanching slope under water
dryslp	1.0	0.1	2.	[-]	critical avalanching slope above water
por	0.4	0.3	0.5	[-]	porosity

Table 5.1 Description of input parameters in *params.txt*

The format of file *params.txt* is that of a simple *.ini* file, where *keyword=value* combinations are given in any order. Lines that do not contain an '=' sign are treated as comment lines. Below, an example is given for a model of the 'minigrid' area at Duck, NC.

```
Grid input

nx      = 140
ny      = 80
dx      = 5.
dy      = 10.
Xori    = 800.
Yori    = 1500.
Alfa    = 180.
depfile = duck-ext.dep

Wave input

Hrms  = 3.
Tm01  = 10.
dir0  = 60.
m     = 10
nt    = 2000
Hmin  = 0.01
Tlong = 80.
gamma = 0.6
alpha = 1.
delta = 0.0
n     =  5.
rho   = 1025
g     = 9.81
thetamin = 0.
thetamax = 60.
dtheta   = 10.
wci      = 0
break    = 2
instat   = 1

Flow input

C=60.
eps=0.1
Umin=0.1
zs0=0.d0
tstart=0
Tint=5.
tstop=2000
CFL=0.2
nuh=0.0

sed input

A        = 2.e-3
Dico     = 1.
D50      = 0.0002
D90      = 0.0003
Rhos     = 2650
Morfac   = 0.0
```

5.2 Output

All relevant variables are written as double precision reals to binary output files, one file per variable, with a name of *<variable>.dat*. The grid is written to a file *xy.dat* and the dimensions are written to *dims.dat.* The output time interval is specified by keyword *tint*.
This format is easy to write and read in both Matlab and Fortran, without additional libraries.

In a later stage output may be provided to a standard such as XMDF, NetCDF, HDF, once exact standards have been agreed within Morphos.

Below, a sample Matlab script is shown illustrating how to read the output.

```
% Read dimensions
fid=fopen('dims.dat','r');
nt=fread(fid,[1],'double')
nx=fread(fid,[1],'double')
ny=fread(fid,[1],'double')
fclose(fid)
%
% Read grid
fixy=fopen('xy.dat','r');
x=fread(fid,[nx+1,ny+1],'double');
y=fread(fid,[nx+1,ny+1],'double');
fclose(fixy)
%
% Open data files
fid=fopen('Hrms.dat','r');
fiz=fopen('zb.dat','r');
fiu=fopen('u.dat','r');
fiv=fopen('v.dat','r');
%
% Open figure
figure(1);
%
% Start time loop
for i=1:nt
    %
    % Read Hrms, zb, u,v
    f=fread(fid,[nx+1,ny+1],'double');
    z=fread(fiz,[nx+1,ny+1],'double');
    if i==1
        z0=z;
    end
    u=fread(fiu,[nx+1,ny+1],'double');
    v=fread(fiv,[nx+1,ny+1],'double');
    % Plot Hrms in left panel
    subplot(121);pcolor(x,y,f);   shading interp; colorbar;
    % Plot sedimentation/erosion in right panel
    subplot(122);pcolor(x,y,z-z0);shading interp; colorbar;
    hold on
    % Add vector plot velocity
    quiver(x,y,u,v,1); hold off;
    title(num2str(i));drawnow
end;
fclose(fid)
fclose(fiz)
fclose(fiu)
fclose(fiv)
```

6 Distribution and maintenance

At this moment, the source code and some test case inputs are available at UNESCO-IHE's collaborative platform, to which a selected group of developers in Delft and at ERDC have access. In addition, first steps have been taken, in the framework of the NOPP-CSTM project, to bring the model under Subversion version management. In the coming period, after experimenting with the structure of the version management tree, we will make this way of working operational.

7 Conclusions and future work

Significant progress has been made over the last year in modelling the dune erosion process: a new model has been devised and coded and a number of validation tests has been carried out. The approach tested so far has been to run in nonstationary mode, where wave-group generated long waves are generated which represent the dominant motion in the swash zone. As these motions are also likely to dominate overwashing processes we are eager to proceed to test cases where overwashing occurs.

A drawback of the modelling approach may be the required resolution, in the order of metres; in 1D simulations this is no problem, in 2DH simulations covering larger domains it may become restrictive, though parallellization can solve much of this problem. Still, it is worth considering alternative 'quick-and-dirty' approaches based on stationary wave and flow modelling combined with an extrapolation method for the actual dune erosion.

Our plans for the coming year are summarized below; of course these are subject to discussions with ERDC staff and Morphos project group, within the budgetary constraints.

- Investigate Steetzel's formulations, Van Rijn's latest method and original D3D "extrapolation method" so that "quick and dirty" runs can be made.
- Include routines to generate omni-directional non-stationary short and longwave boundary conditions which require specification of a 2D spectrum as input.
- Establish coupling with larger-domain short wave models (typically frequency-domain wave action models such as ST/Wave, TS/Wave or SWAN) to provide these 2D input spectra.
- Implement slowly-varying wave parameters to reflect storm history
- Implement (slowly) time-varying water levels to represent surge and tide effects, both on the seaward and tidal inlet side of the domain
- Include Q3D description of flow cf Reniers et al, 2004b.
- Perform validation for all relevant cases presently in Delft Hydraulics test bed format and couple Xbeach to this test bed.
- Review formulations of wave impact contributions by short waves and long waves (e.g. Overton and Fisher).
- Future tests for testbed:
    - New dune erosion test from Deltaflume (available)
    - Berm test (available)
    - Scheveningen berm test in Deltaflume
    - Oregon test with overwash.
    - 2D tests field tests: Cape Hatteras (Isabel), contact persons: Brad Johnson, Abby Sallenger and Florida cases, contact person Dave Froehlich.
- Training on the use of Xbeach and Delft3D morphology at Vicksburg (proposed).
- Attend Morphos workshop at Oahu, Hawaii in Fall 2007.

References

Abbott, M.B., 1979. Computational Hydraulics. Elements of the theory of free surface flow. Pitman Publishing, London, 324 pp.

Arcilla, A.S., Roelvink, J.A., O'Connor, B.A. Reniers, A., and Jimenez. J.A. (1994). The Delta Flume '93 Experiment. Coastal Dynamics '94. Arcilla, Stive and Kraus (eds), ASCE, New York, pp. 488-502.

Carrier, G.F. and H.P. Greenspan (1958). Water waves of finite amplitude on a sloping beach. J. Fluid Mech., Vol. 4, pp. 97-109.

Galapatti, R., 1983. A depth integrated model for suspended transport. Report 83-7, Communications on Hydraulics, Department of Civil Engineering, Delft University of Technology.

Reniers, A.J.H.M., J.A. Roelvink and E.B. Thornton. (2004). Morphodynamic modelling of an embayed beach under wave group forcing. J. of Geophysical Res. , VOL. 109, C01030, doi:10.1029/2002JC001586, 2004

Reniers, A.J.H.M., E.B. Thornton, T. Stanton and J.A. Roelvink. (2004b) Vertical flow structure during Sandy Duck: Observations and Modeling. Coastal Engineering, Volume 51, Issue 3, May 2004, Pages 237-260

Roelvink, J.A. (1993a) Dissipation in random wave groups incident on a beach. Coastal Eng., 19 (1993) pp. 127-150.

Roelvink, J.A. (1993b) Surf beat and its effect on cross-shore profiles. Ph.D. thesis, Delft University of Technology.

Soulsby, R. (1997). Dynamics of marine sands. Thomas Telford Publications, London, ISBN 0 7277 2584 X.

Stelling GS, Duinmeijer SPA. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. International Journal for Numerical Methods in Fluids. 2003; 43:1329-1354.

Van Dongeren, A.R. and I.A. Svendsen (1997). An Absorbing-Generating Boundary condition for Shallow Water Models. J. of Waterways, Ports, Coastal and Ocean Engineering, vol. 123, no. 6, pp. 303-313.

Verboom, G.K., G.S. Stelling and M.J. Officier (1981). Boundary conditions for the shallow water equations. In: Abbott, M.B. and J.A. Cung, eds. Engineering Applications of Computational Hydraulics, vol. 1., pp. 230-262.